Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

A Comprehensive Examination of Risk Management Practices Throughout the Software Development Life Cycle (SDLC): A Systematic Literature Review

Zahrina Aulia Adriani[1], Teguh Raharjo[2], Ni Wayan Trisnawaty[3]
zahrina.aulia@ui.ac.id, teguhr2000@gmail.com, ni.wayan05@ui.ac.id
[1,2,3]Magister Teknologi Informasi,Fakultas Ilmu Komputer, Universitas Indonesia

Article Information

Submitted : 12 May 2024
Reviewed : 16 May 2024
Accepted : 15 Jun 2024

Keywords

Risk Management, Software Development Life Cycle (SLDC), Systematic Literature Review (SLR)

Abstract

Risk management in the software development lifecycle (SDLC) is a continuous process that addresses risks throughout a system's lifecycle, including acquisition, development, maintenance, or operation. Despite its importance, ineffective risk management practices can lead to project failures, impacting organizations financially and reputationally. Therefore, there is a need for a systematic understanding of risk management practices in SDLC. This study conducts a Systematic Literature Review (SLR) related to risk management activities performed by previous research during the SDLC. The SLR method combines Kitchenham with the toll-gate method to select literature for use. This SLR aims to investigate activities in traditional waterfall and agile development processes, which will be mapped into risk management activities in SDLC according to ISO 16085:202. Additionally, the review highlights the challenges encountered in implementing risk management in the SDLC process, including project complexity, adherence to policies and standards, lack of communication, lack of resources, and organizational culture.

A. Introduction

Software development follows a structured path, commencing with planning and progressing through design, development, testing, implementation, maintenance, and support, collectively called the Software Development Life Cycle (SDLC) [1]. Throughout this process, software development projects are inherently prone to risks that can significantly impact their success, quality, and profitability [2]. Effective risk management is crucial to mitigate these risks, ensuring that software development projects are completed on time, within budget, and meet the required quality standards [3].

The Standish Group's Annual CHAOS 2020 report reveals alarming statistics, indicating that 66% of technology projects globally, drawn from an analysis of 50,000 endeavors, culminated in partial or complete failure [4]. Notably, even small-scale software projects have a one-in-ten chance of failure, underscoring the pervasive nature of this issue [5]. As software systems evolve into large-scale entities characterized by heightened complexity, the accompanying growth in project size and intricacy amplifies associated risks [6]. Consequently, in the contemporary landscape, managing these risks is imperative to navigate challenges and bolster the likelihood of project success [7].

According to the Project Management Body Of Knowledge (PMBOK), risk can be defined as an event or series of uncertain events and, if they occur, will have a negative or positive effect on one or more project objectives [8]. Risk can also be defined as the possibility of loss arising when a threat exposes a vulnerability [9]. Risks identified in a company's assets can bring tangible and intangible value. Tangible value relates to actual costs, such as lost revenue and repair costs. In contrast, Intangible value refers to worth that extends beyond monetary measures, encompassing elements like customer trust, potential future losses, and the sway customers have over others [10].

Based on ISO 12207 (a standard related to the software development life cycle), a software product or service development goal may encompass various dimensions, including financial viability, health outcomes, security measures, and environmental impact [11]. Risks can arise at multiple levels of an organization and can be caused by internal and external factors [12]. These factors subsequently influence the likelihood of risk occurrence and its impact on business objectives. In the context of SDLC, risks can arise at the project, product, and process levels [13].

Suppose there is no mitigation against the risks identified during the SDLC process. In that case, the company will incur losses related to product quality, increased production costs, added time to project completion, and failure to meet predetermined timelines [17]. Identifying and tracking risks will help improve project success and achieve quality software [18]. In this regard, risk management plays a crucial role in identifying risks [11].

Despite the importance of risk management in software development, many projects struggle with ineffective risk management practices, leading to delays, cost overruns, and poor quality [19]. The lack of a comprehensive risk management approach can result in project failures, which can have significant financial and reputational consequences for organizations [20]. Therefore, there is a growing need for a systematic understanding of risk management practices

throughout the SDLC to identify best practices, challenges, and areas for improvement.

Several studies have discussed risk management in software development life cycles, including those conducted by research [21] and [22]. Research [21] evaluated the application of risk management in various software development methodologies, while research [22] conducted a literature review on risks related to traditional and agile software development. However, both studies used the framework risk management ISO 31000:2018, a general organizational risk management standard. This standard does not specifically regulate risk management in software development [23]. To address risk management in SDLC, one can use the more specific standard ISO 16085:2021 for software development [24].

In this study, we will conduct a systematic literature review (SLR) on literature related to risk management conducted during the SDLC. The results of the SLR will be mapped to SDLC activities with occurring risk management activities, challenges faced, and recommendations for overcoming constraints encountered during the implementation of risk management in the SDLC. Risk management activities will be based on ISO 16085:2021, which discusses a comprehensive and structured framework for managing risks in system and software engineering projects [24].

This research is structured as follows: Part 1 explains the introduction, Part 2 describes the research methodology conducted, Part 4 explains the results and discussion, and Part 5 presents the research conclusions.

Risk Management

Risk management is a systematic process conducted to identify, assess, and prioritize risks associated with every asset or project used to reduce their impact on operational activities and organizational objectives [25]. Based on ISO 16085:2021, the risk management process conducted in the SDLC consists of 7 stages visualized in Figure 1. The following are the stages that occur:

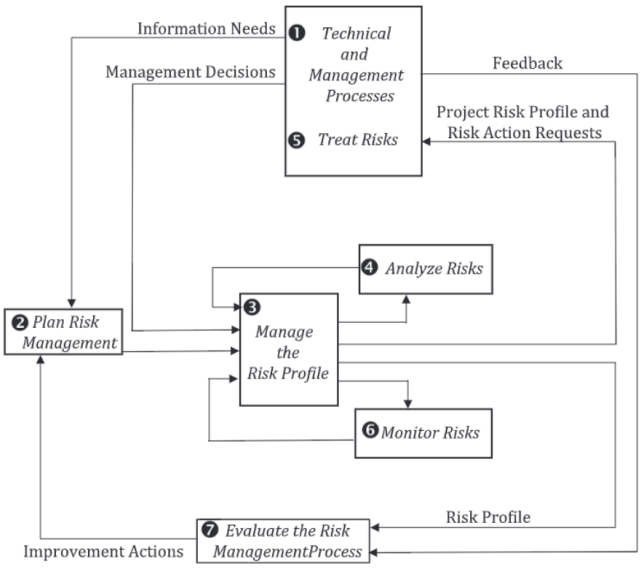

Figure 1. Risk Management Process Model

1) Technical and Management Processes

This process involves stakeholders used to gather information about the scope and boundaries, both internally and externally, related to risks within the organization. The organization should identify internal and external elements that may influence the objectives and outcomes of risk management. Internal factors may involve the organization's culture, structure, and assets, while external factors may include the regulatory environment, economic conditions, and market trends [26], [27].

2) Plan Risk Management

Planning risk management consists of two activities: defining the risk management strategy and defining and recording the context of the risk management process. A risk management strategy is conducted to determine the objectives and scope of risk management activities to be carried out. This strategy includes risk tolerance questions that explain the project's attitude toward risk-taking and influence the activities and tasks in the project management process [26], [28]. The next activity is to define and record the context of the risk management process, which includes a list of stakeholders, descriptions of relevant stakeholders, risk categories, descriptions of technical and managerial objectives, assumptions and constraints, and relevant information that can influence risk analysis and treatment [26], [28].

3) Manage the Risk Profile

In managing risk profiles, the activities carried out are defining and recording the risk thresholds and conditions and establishing and maintaining the risk profile. Defining and recording the risk thresholds and conditions are done to determine the level of risk exposure handled or accepted by the development project team. By establishing risk thresholds and conditions, actions to mitigate these risks are more measurable, allowing the organization to know when to accept, monitor, or address risks according to the risk threshold values obtained [26], [29]. Furthermore, establishing and maintaining the risk profile is conducted to ensure that current and historical risks that occur during the project remain consistent. By understanding the risk profile well, organizations can take more effective actions to reduce the impact of risks and leverage potential opportunities that may arise [26], [29].

4) Analyze Risks

Risk analysis is conducted to understand better the risks that may affect a project, activity, or organization. In risk analysis activities, several tasks are performed, namely: identifying risks in the categories described in the risk management context, estimating the likelihood of occurrence and consequence of each identified risk, evaluating each risk against its risk threshold, defining and recording recommended treatment strategies and measures [26], [30].

5) Treat Risks

Treat risk is systematically applying policies, procedures, and practices to modify risk. Treating risk aims to minimize negative consequences and their

likelihood while potentially increasing the possibility of positive effects. These modifications can include various actions to manage and control the level of risk. Activities to address identified risks include identifying recommended alternatives for risk treatment, implementing risk treatment alternatives, monitoring high-priority risks, and coordinating management actions for selected risk treatments [26], [30].

6) Monitor Risks

Monitoring should be carried out by the organization to demonstrate the success of risk management plans, strategies, and management systems that have been implemented to manage risks effectively. There are several activities conducted in risk monitoring, namely, continually monitoring the risk management context, implementing and monitoring measures to evaluate the effectiveness of risk treatments, and continuously monitoring for the emergence of new risks and sources throughout the life cycle [26], [31].

7) Evaluate the Risk Management Process

The evaluation of the risk management process is conducted to measure and assess the effectiveness of the process in identifying, analyzing, managing, and mitigating risks that may affect a project, activity, or organization. Activities carried out during risk evaluation include analyzing recurring issues, problems, and risks over time, identifying lessons learned, and improving the risk management process [26], [32].

B. Research Method

This section will explain the method used to conduct the Systematic Literature Review (SLR) using Kitchenham to address the research questions. Systematic Literature Review is a literature review method used to identify, assess, and interpret all findings in a study to answer pre-defined research questions [33]. SLR has three main stages: planning, implemention, and reporting the results of the SLR, as described in Table 1. The following is a detailed explanation of each step of the SLR conducted:

Table 1. SLR Phase

Phases	Steps
Planning SLR	Search Strategy
	Inclusion and Exclusion
	Quality Assessment
Implementation SLR	Primary Study Selection
	Data Extraction
	Data Synthesis
Reporting SLR	Documenting the extracted result

Planning SLR

Planning for the SLR is the first step in preparing and conducting the SLR. This process is done to ensure the success and validity of the review. The following is an explanation of the planning stages of the SLR conducted:

1) Search Strategy

Research questions are primary in determining search and analysis strategies when conducting the SLR. We have identified the research questions in this study:

RQ1: What risk management activities have been addressed by the related studies?

RQ2: What challenges are faced when implementing risk management in the SDLC?

To search, we used a logical connector search string [34]. In utilizing logical connectors, the OR logical operator is used for alternative terms, and AND combines these terms. With the specified keywords, the following is the search strategy that will be used in this research:

(("implementation" OR "challenge") AND ("risk management" OR "risk mitigation") AND ("software development life cycle" OR "software engineering life cycle") OR "software risk"))

2) Inclusion and Exclusion Criteria

The inclusion and exclusion criteria will be used to select the literature. Inclusion criteria select relevant literature studies suitable for the research questions. In contrast, exclusion criteria eliminate studies irrelevant to the research questions [33]. Table 3 describes the inclusion and exclusion criteria used in this research.

Table 2. Inclusion and Exclusion Criteria

Code	Inclusion Criteria	Code	Exclusion Criteria
IN1	The literature published between 2018 and 2023	EX1	Literature related to risk management that is not specific to SDLC
IN2	The literature is published in journals and conferences	EX2	Literature that is unrelated to risk management in SDLC
IN3	The literature is written in English	EX3	Literature discussing types of risks
IN4	The literature must provide answers to the research questions	EX4	Literature from books, magazines, and blogs
		EX5	Literature that does not answer the research questions
		EX6	Literature that is not accessible with full-text
		EX7	Systematic Literature Review literature
		EX8	Duplicate literature

Implementation SLR

We extract all data from publications using the inclusion and exclusion criteria and the research questions for study quality. The toll-gate method is employed to refine the research articles identified during the literature collection process, ensuring the selection of high-quality literature. [35], [36]. Tabel 3 shows five steps associated with toll-gate method.

The toll-gate method is performed after we select literature based on inclusion and exclusion criteria, resulting in 412 works of literature related to the research topic. After conducting these 5 phases, we found 28 literature considered as primary studies. Finally, quality evaluation is applied to the literature selected by

the toll-gate method. Table 5 provides the results of the election conducted using the toll-gate method.

Table 3. Toll-gate steps

Toll-gate method
Step 1: Utilizing search terms to discover relevant articles
Step 2: Assessing articles based on their title and abstract
Step 3: Assessing articles based on introduction and conclusion
Step 4: Assessing articles based on full-text reading
Step 5: Compiling the final primary literature for SLR using predetermined criteria for assessing study quality.

Table 4. SLR Process Result

Electronic Database	1	2	3	4	5
IEEE	167	48	30	20	
Science Direct	57	19	10	17	
Emerald Insight	35	21	11	4	44
ACM	86	23	13	9	
Total	345	111	64	50	

Reporting SLR

The reporting phase of the SLR is a crucial stage that involves a comprehensive analysis of insights gained from previous steps. In this stage, findings will be gathered and documented [37]. We will analyze the risk management processes undertaken and the challenges faced in the Software Development Life Cycle (SDLC) based on the literature studies conducted.

C. Result and Discussion

After completing the predetermined research steps outlined in the methodology section, this section will present and discuss the results obtained from the SLR process.

Literature Demography and Visualization

In this section, we will explain the demographics of the selected journals. Figure 4 will illustrate the distribution of journals related to the research topic from 2018 to 2023. Based on the graph results in Figure 4, research on risk management in the SDLC was most prolific in 2022, with nine literatures.

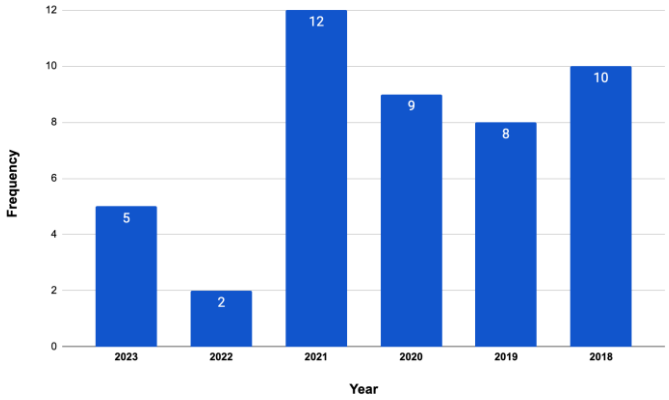

Figure 2. Journal Distribution 2018-2023

After identifying the journal distribution, we identified keywords based on the titles and abstracts available in the journals. Figure 5 displays the network of keywords frequently appearing in the primary studies. This keyword network was created using the VOSviewer application.

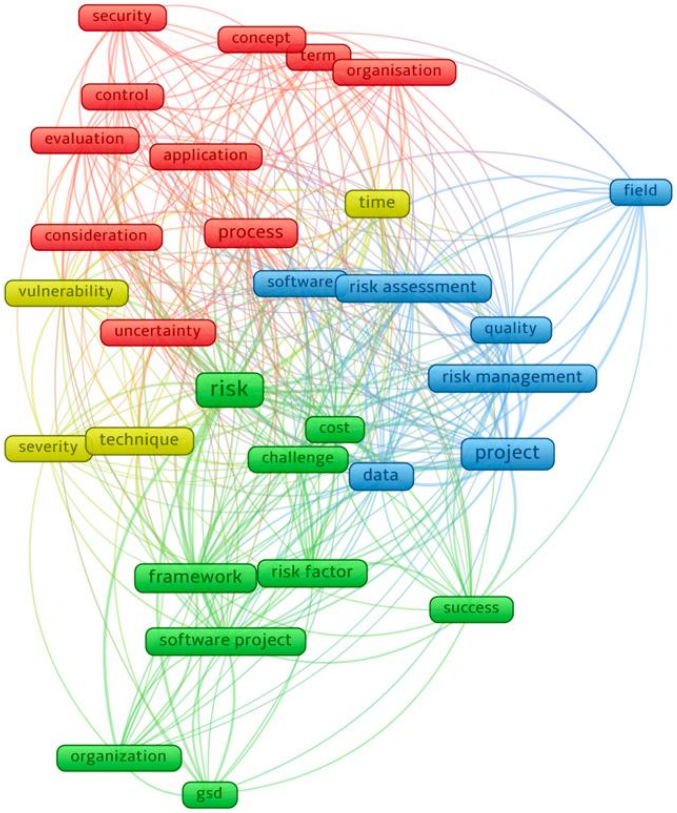

Figure 3. Visualization of Selected Literature Keywords

From the visualization of the keyword network shown in Figure 5, many keywords indicate essential elements in the stages of the SDLC, such as time, process, cost, security, and data, which can experience vulnerabilities and bring risks during the software project development process. Regarding risk management, the most dominant based on keywords is risk assessment during the SDLC. To examine in more detail the risk management processes conducted in the literature, we will map the risk management activities, challenges faced, and recommendations provided by the literature during the implementation of risk management in the SDLC in the next section.

Risk Management Activities

Based on ISO 16085:2021, the risk management activities conducted to control risks that occur during the SDLC consist of 6 stages: technical and management processes, plan risk management, manage the risk profile, analyze risks, treat risks, monitor risks, and evaluate the risk management process [24]. Table 6 shows the mapping of risk management activities to the SDLC process, which is both traditional (Waterfall) and agile (Scrum). This process is done to

identify activities conducted parallel to the two approaches. The following is a more detailed explanation of the mapping performed:

Table 5. Mapping SDLC - Risk Management Activities

Risk Management	Traditional Development	Agile Development
Technical and Management Processes	Requirement Analysis	Sprint Planning, Grooming Backlog
Plan Risk Management	Requirement Analysis	Sprint Planning, Grooming Backlog
Manage Risk Profile	System Design	Sprint Planning, Grooming Backlog
Analyze Risk	Implementation	Sprint Planning, Grooming Backlog
Treat Risk	Testing	Daily Scrum
Monitor Risk	Deployment	Daily Scrum
Evaluate Risk Management	Maintenance	Sprint Review dan Sprint Retrospective

1) Traditional Development

The first stage in the waterfall process, which is traditional development, is defining requirements, which involves understanding the business context and the scope of software development [38]. Additionally, developers elicit requirements from stakeholders regarding the software's functionality (functional and non-functional) and features [39]. At the same time, risk management activities are also conducted in technical and management processes and plan risk management [40]. Technical and management processes are carried out while identifying the scope of software development. Defining the context (determining scope and boundaries) of a software development project can assist in managing project resources, timelines, and the risks involved [41].

The second stage is the design phase. In the design phase, the project team will use the results of the requirements identification and logical model generated from the previous process as inputs in designing the architecture that supports software design [42]. In this process, the project team also analyzes the impact and likelihood of project development risks, such as compiling the existing risk profile in the risk profile activity [43].

The third stage is implementation. During this process, the development team will use the requirements specifications and design documents prepared in the previous stage as a guide for coding [44]. This process aligns with the risk analysis process that occurs in risk management. In the implementation of SDLC, risk identification may include risks related to system failures, resource shortages, or errors in coding [13]. Additionally, estimating and evaluating potential risks are carried out to determine which risks are most likely to occur in the system development process and should be prioritized for mitigation [45].

The fourth stage involves testing or evaluating the developed product. Testing will be conducted to ensure that there are no bugs and that end-user experiences are not disrupted [46]. During the testing phase, developers will carefully inspect their software, noting any bugs or defects that must be tracked, fixed, and retested. This process is also done in conjunction with the risk

management process, namely, treating risks by taking various actions such as risk avoidance, risk mitigation, risk transfer, and risk acceptance [47], [48].

After ensuring that the software meets the requirements through the testing process, the next stage is deployment [49]. The deployment phase will involve installing the software by creating installation guides, operating systems, and functions accessible to end-users [50]. This process also involves monitoring for emerging risks, such as critical bugs, or identifying potential issues or arising risks during the deployment process, which can then be considered new risks [51].

The final stage in the software development process using traditional development is maintenance. The maintenance phase involves maintaining the software to ensure performance aligns with issues or risks detected after use, adding, or changing features based on user feedback [52]. During maintenance, the risks found due to adding and altering software features will be evaluated. This process enables the team to make informed decisions and prioritize feature changes according to the associated risks [21].

2) Agile Development

Sprint Planning is the first stage in the Scrum process, which is one of the agile development approaches. In this stage, the team forecasts the work to be completed in the next sprint. Sprint planning sets the sprint goals, details the work, estimates the time required, and creates the Sprint Backlog [53]. The next activity is called Grooming Backlog. Grooming Backlog is a periodic activity where the team updates, prioritizes and elaborates on items in the Product Backlog. The aim is to ensure that the items in the Product Backlog are ready to be included in the Sprint Backlog during Sprint Planning [54]. Grooming Backlog also helps the team clarify the details and estimates of work and adjust the priority of items based on changing business needs [55]. This procedure aligns with several activities in risk management, such as technical and management processes, plan risk management, manage risk profile, and analyze risk. This event happened because the planning and software design process carried out in Scrum occurs in every sprint [56].

The next step is the Daily Scrum. The Daily Scrum is a brief daily meeting conducted by the development team within the Scrum framework to synchronize the team, coordinate work, and identify any obstacles encountered during development [57], [58]. Risk management activities in this process include monitoring and treating risks [58]. In the risk monitoring process, the team will monitor anything related to potential new risks that emerge during development [59]. Risk monitoring also tracks the progress of implementing previously planned risk mitigation actions. As for the risk treatment activity, it involves taking necessary mitigation actions to reduce the negative impact of the risks identified [60].

The process that concludes a sprint is the sprint review and sprint retrospective. In the sprint review process, the development team and stakeholders engage to review the work results achieved during the sprint, ensuring business needs and feedback are addressed and making decisions for the next steps [61]. Subsequently, for the sprint retrospective, the team will review the overall performance throughout the sprint. This process is done to gather lessons learned to be used as opportunities for improvement in the next sprint to provide

more excellent value to the developed product related to overall performance, and issues or risks arising from both processes are used to evaluate risk management strategies and update mitigation plans [62], [63]. Integrating these processes enables the development team to proactively manage risks and enhance project success [64].

Challenges in Implementing Risk Management

During the implementation of risk management in the SDLC, as discussed in the literature, several challenges were encountered. Here are the challenges identified:

1) Project Complexity

The first challenge is project complexity. Complex software projects involve intricate features, abundant integrations, unfamiliar technologies, large teams with varied skills, and strict time and budget constraints. This complexity impacts risk management in software development [65]. Firstly, there is difficulty in identifying all risks because complex projects have more potential failure points that may be overlooked [66]. Additionally, the interconnected nature of complex projects makes it challenging to predict how risks in one area may affect others. Furthermore, complex projects require more comprehensive and flexible mitigation plans to address unforeseen issues [65].

2) Policies and Standards

The next challenge is related to policies and standards. Policies and standards provide a framework for identifying, assessing, mitigating, and monitoring risks throughout the development [67]. Policies referred to are the guidelines for risk management in the development process. This document contains roles and responsibilities for risk management activities (identification, assessment, mitigation, monitoring), risk management processes to be followed throughout the SDLC, and the level of risk tolerance (the level of risk that the organization can accept) [84]. Standards are defined as more specific guidelines that provide practical details on implementing risk management policies [68].

In this domain, the challenges usually experienced include stakeholders' lack of understanding and awareness of the policies and standards set for the project, leading to errors in interpreting the applicable provisions [69]. Additionally, there's often a lack of support and commitment from senior management to implement policies and standards consistently. Inconsistencies and fragmentation of policies and standards across various departments also pose challenges and difficulties in integrating them with existing risk management frameworks [70].

3) Lack of Communication

The next challenge relates to insufficient communication in software development risk management. During the risk identification stage, there is often limited communication, causing team members to not fully understand the project's objectives, dependencies, and potential challenges [71]. Consequently, many risks go unidentified due to inadequate communication. Without open communication among stakeholders, conflicts may arise in risk selection, leading

to inaccurate risk assessments [72]. Disruptions in team communication can also delay conveying information about emerging risks, rendering applied mitigation strategies ineffective [73]. Limited communication may also prevent the team from being aware of changes in project scope, deadlines, or technology, potentially resulting in neglect of how these changes affect existing risks [74].

4) Lack of Resources

Resource shortages in the software development lifecycle can pose challenges in risk management, resulting in various issues, such as increased unidentified risks due to limited time and team skills [75]. Resource shortages can also limit the development of comprehensive contingency plans, and without adequate resources, the team will struggle to assess risks during development [76]. Resource constraints can also hinder regular risk monitoring, causing planned mitigation strategies not to proceed as planned [77].

5) Organization Culture

Organizational culture can also be a significant barrier to effective risk management implementation. The first challenge in this domain is the lack of awareness and understanding of individuals' roles and responsibilities in risk management [78]. This challenge is often caused by senior management's lack of commitment to allocate resources and time for risk management activities related to the software development process [79]. Another challenge is a culture that blames team members who report risks out of fear of being held accountable, which can lead to overlooked risks and delayed mitigation efforts [80]. Additionally, reactive approaches such as waiting for problems to arise before addressing them are common in organizational culture issues, leading to rushed decisions and ineffective risk mitigation strategies [81].

D. Conclusion

Based on the literature selected from 2018-2023, the study concludes that risk management in software development is a crucial factor in ensuring project success. 44 relevant pieces of literature meeting the criteria for journal quality were identified for review. Through keyword visualization, it was observed that five words are associated with essential elements in the stages of the SDLC: time, process, cost, security, and data. These elements are prone to vulnerabilities and risks during the software project development process. Both the traditional waterfall process and agile development approaches, such as Scrum, integrate risk management activities throughout their stages. However, implementing risk management in software development is not without challenges. These challenges include project complexity, adherence to policies and standards, communication gaps, resource constraints, and organizational culture. These hurdles may hinder the identification, assessment, mitigation, and monitoring of risks, ultimately affecting the success of software development projects. Thus, software development teams must recognize and address these challenges proactively to ensure effective risk management and project success. For future research, it is recommended that scholars focus on devising strategies to tackle the challenges associated with implementing risk management in software development.

E. References

[1] J. T. Marchewka, *Information technology project management: Providing measurable organizational value*. John Wiley & Sons, 2016.

[2] O. Skakalina and A. Kapiton, "Identification And Management Of Risks In The Project Management Of The Development Of Software Products," vol. 1, no. 71, pp. 145–149, 2023.

[3] M. H. Zahedi, A. R. Kashanaki, and E. Farahani, "Risk management framework in Agile software development methodology.," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 4, 2023.

[4] The Standish Group, "The Standish Group 2020 Chaos Report — Beyond Infinity," 2020.

[5] M. Jørgensen, "A survey on the characteristics of projects with success in delivering client benefits," *Inf. Softw. Technol.*, vol. 78, pp. 83–94, 2016, [Online]. Available: https://api.semanticscholar.org/CorpusID:12951836

[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] A. Sousa, J. P. Faria, and J. Mendes-Moreira, "An analysis of the state of the art of machine learning for risk assessment in software projects," in *Proceedings of the 33rd International Conference on Software Engineering and Knowledge Engineering, SEKE*, 2021, pp. 1–10.

[8] PMI, *PMBOK,6th edition*. 2017.

[9] D. Gibson and A. Igonor, *Managing risk in information systems*. Jones & Bartlett Learning, 2020.

[10] H. Pandya and A. Jain, "Relationship between intangible assets and firm value: a study of selected Indian companies," *ZENITH Int. J. Bus. Econ. Manag. Res.*, vol. 5, pp. 132–148, 2015, [Online]. Available: https://api.semanticscholar.org/CorpusID:155469513

[11] I. S. O. ISO, "IEC 12207 Systems and software engineering-software life cycle processes," *Int. Organ. Stand. Geneva*, 2008.

[12] A. Q. Adeleke *et al.*, "The influence of organizational external factors on construction risk management among Nigerian construction companies," *Saf. Health Work*, vol. 9, no. 1, pp. 115–124, 2018.

[13] A. Hannah, A. Kamal, C. Chuah, Y. Yen, G. Jia, and Hui, "Risk Assessment, Threat Modeling and Security Testing in SDLC," *arXiv.org*, 2020.

[14] C. Kumar and D. K. Yadav, "A Probabilistic Software Risk Assessment and Estimation Model for Software Projects," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:58105384

[15] H. A. Dahalan, "Contractual requirement list for project data management system and software testing activities," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:107429701

[16] R. K. Bhujang and S. V Dean, "Propagation of Risk across the Phases of Software Development," in *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, 2018, pp. 508–512. doi: 10.1109/I-SMAC.2018.8653647.

[17] M. Hammad, A. Abbasi, R. K. Chakrabortty, and M. J. Ryan, "Predicting the

critical path changes using sensitivity analysis: a delay analysis approach," *Int. J. Manag. Proj. Bus.*, vol. 13, no. 5, pp. 1097–1119, Jan. 2020, doi: 10.1108/IJMPB-07-2019-0184.

[18] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying software project risks: An international Delphi study," *J. Manag. Inf. Syst.*, vol. 17, no. 4, pp. 5–36, 2001.

[19] A. Sols, "A Comprehensive Approach to Dynamic Project Risk Management," *Eng. Manag. J.*, vol. 30, pp. 128–140, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:117330704

[20] T. Dingsøyr and Y. Petit, "Managing layers of risk: Uncertainty in large development programs combining agile software development and traditional project management," *ArXiv*, vol. abs/2103.09034, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:232240449

[21] P. Chemweno, L. Pintelon, P. N. Muchiri, and A. Van Horenbeek, "Risk assessment methodologies in maintenance decision making: A review of dependability modelling approaches," *Reliab. Eng. Syst. Saf.*, vol. 173, pp. 64–77, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:3798542

[22] M. Pilliang and M. Munawar, "Risk Management in Software Development Projects: A Systematic Literature Review," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 8, no. 2, 2022.

[23] ISO, "ISO 31000: 2018 Risk Management—Guidelines." International Organization for Standardization Geneva, 2018.

[24] ISO, "ISO/IEC/IEEE 16085:2021(en), Systems and software engineering — Life cycle processes — Risk management," 2018. https://www.iso.org/obp/ui/#iso:std:iso-iec-ieee:16085:ed-1:v1:en (accessed Jan. 29, 2024).

[25] "Risk Management: Is It Needed?," in *Adaptive Security and Cyber Assurance for Risk-Based Decision Making*, Hershey, PA, USA: IGI Global, 2023, pp. 24–41. doi: 10.4018/978-1-6684-7766-3.ch002.

[26] ISO/IEC/IEEE 15288, "International Standard ISO/IEC/IEEE 15288 Systems and Software engineering - System life cycle processes," *ISO*, vol. 17, no. 1, p. 108, 2015.

[27] F. Ackermann, S. Howick, J. Quigley, L. Walls, and T. Houghton, "Managing projects in an uncertain world : engaging stakeholders, and building a systemic view of risk," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:109893921

[28] A. Fissore, P. Cova, and C. De Matteo, "Scenario planning as an element of strategic risk management," *Impresa Progett. - Electron. J. Manag.*, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:169773625

[29] I. Rybka and E. Bondar-Nowakowska, "Planning of the Risk Handling Methods Related to Alterations to Project Documentation," *Procedia Eng.*, vol. 57, pp. 952–957, 2013, [Online]. Available: https://api.semanticscholar.org/CorpusID:110663319

[30] A. T. Bahill and A. M. Madni, "Risk Analysis and Management," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:168383441

[31] N. G. Mutlu and S. Altuntas, "Developing an integrated conceptual framework

for monitoring and controlling risks related to occupational health and safety," *J. Eng. Res.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:244933913

[32] T. Karkoszka, "Technological Risk in the Context of Process Approach and Risk Assessment," *Syst. Saf. Hum. - Tech. Facil. - Environ.*, vol. 3, pp. 312–319, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:235486589

[33] B. Kitchenham and S. M. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007, Accessed: Mar. 19, 2023. [Online]. Available: https://www.researchgate.net/publication/302924724

[34] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *J. Syst. Softw.*, vol. 80, no. 4, pp. 571–583, 2007.

[35] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Inf. Softw. Technol.*, vol. 51, no. 6, pp. 957–976, 2009.

[36] F. Matloob *et al.*, "Software defect prediction using ensemble learning: A systematic literature review," *IEEE Access*, vol. 9, pp. 98754–98771, 2021.

[37] G. Tebes, D. Peppino, P. Becker, and L. A. O. Santos, "Enhancing the Process Specification for Systematic Literature Reviews," 2019.

[38] S. H. Guruwada, "Understanding Requirement Analysis Phase," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 474–476, Apr. 2021, doi: 10.32628/CSEIT217293.

[39] P. Haindl, R. Plösch, and C. Körner, "Tailoring Stakeholder Interests to Task-Oriented Functional Requirements," *arXiv.org*, 2022, doi: 10.13140/RG.2.2.16632.16641.

[40] M. Karminska-Bielobrova and N. Shmatko, "RISK-MANAGEMENT AS AN ASPECT OF OPERATIONAL MANAGEMENT," *Bull. Natl. Tech. Univ.* , no. 1, pp. 36–40, Feb. 2021, doi: 10.20998/2519-4461.2021.1.36.

[41] I. U. Hassan and S. Asghar, "A Framework of Software Project Scope Definition Elements: An ISM-DEMATEL Approach," *IEEE Access*, vol. 9, pp. 26839–26870, 2021, doi: 10.1109/ACCESS.2021.3057099.

[42] S. P. Shankar, H. Agrawal, and E. Naresh, "A Survey on Different Approaches to Automating the Design Phase in the Software Development Life Cycle," *Adv. Comput. Electr. Eng.*, pp. 350–372, Apr. 2020, doi: 10.4018/978-1-7998-2772-6.CH018.

[43] H. Soroka-Potrzebna, "Analysis of the risk Identification Stage in Project Management," *Proc. The 3rd Int. Conf. Res. Manag. Econ.*, Jun. 2020, doi: 10.33422/3RD.IMECONF.2020.09.195.

[44] M. A. Adeagbo, J. E. T. Akinsola, A. A. Awoseyi, and F. Kasali, "Project Implementation Decision Using Software Development Life Cycle Models: A Comparative Approach," *J. Comput. Sci. Its Appl.*, vol. 28, no. 1, Sep. 2021, doi: 10.4314/JCSIA.V28I1.10.

[45] R. Oltean, A.-C. Zglobiu, and M. Rus, "Risk Mitigation in Project Management Theoretical Issues and Case Study," *Sci. Bull. Politeh. Univ. Timișoara Trans. Eng. Manag.*, vol. 3, no. 1, pp. 14–29, Apr. 2023, doi: 10.59168/MOEX9694.

[46] A. O. Alsayed and A. L. Bilgrami, "IMPROVING SOFTWARE QUALITY

MANAGEMENT: TESTING, REVIEW, INSPECTION AND WALKTHROUGH," 2018.

[47] E. S. De Oliveira, J. M. P. Neves, A. F. Da Cruz, and E. C. Bezerra, "Work Product Review Process Applied to Test Cases Review for Software Testing," *Brazilian Symp. Softw. Qual.*, pp. 274–280, Nov. 2023, doi: 10.1145/3629479.3629501.

[48] K. Tahera, D. C. Wynn, C. Earl, and C. M. Eckert, "Testing in the incremental design and development of complex products," *Res. Eng. Des.*, vol. 30, no. 2, pp. 291–316, Apr. 2019, doi: 10.1007/S00163-018-0295-6.

[49] O. Lopuha, S. Tsiutsiura, O. Poplavskyi, O. Lysytsin, O. Bondar, and P. Kruk, "Test Design Methodology for Software Verification," *2023 IEEE Int. Conf. Smart Inf. Syst. Technol.*, pp. 241–245, 2023, doi: 10.1109/SIST58284.2023.10223573.

[50] R. Sharma and R. Dadhich, "Analyzing CMMI RSKM with small software industries at level-1," *J. Discret. Math. Sci. Cryptogr.*, vol. 23, no. 1, pp. 249–261, Jan. 2020, doi: 10.1080/09720529.2020.1721888.

[51] R. L. Dillon, G. A. Klein, and E. W. Rogers, "A Monitoring and Warning Framework for Risks," *IEEE Aerosp. Conf.*, vol. 2021-March, Mar. 2021, doi: 10.1109/AERO50100.2021.9438256.

[52] S. Umudova, "ANALYSIS OF SOFTWARE MAINTENANCE PHASES," 2019.

[53] K. V. Melnyk, V. N. Hlushko, and N. V. Borysova, "DECISION SUPPORT TECHNOLOGY FOR SPRINT PLANNING," *Radio Electron. Comput. Sci. Control*, vol. 0, no. 1, pp. 135–145, May 2020, doi: 10.15588/1607-3274-2020-1-14.

[54] A. E. Babiker, A. Mahmoud, and A. Abdalrahman, "Sprint Backlog Estimating and Planning Using Planning Poker Technique in Agile Scrum Framework," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 8, no. 5, p. 109, May 2018, doi: 10.23956/IJARCSSE.V8I5.686.

[55] T. Kravchenko, S. Bruskin, D. V Isaev, and E. V Kuznetsova, "Prioritization of IT Product Backlog Items Using Decision Support Systems," *Inf. Technol.*, vol. 26, pp. 631–640, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:228870544

[56] P. Spagnoletti, N. Kazemargi, and A. Prencipe, "Agile Practices and Organizational Agility in Software Ecosystems," *IEEE Trans. Eng. Manag.*, vol. PP, pp. 1–14, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:244195061

[57] Y. Kaur and S. Singh, "Risk Mitigation Planning, Implementation, and Progress Monitoring: Risk Mitigation," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:134147329

[58] N. Hasib, S. W. A. Rizvi, and V. Katiyar, "Risk Mitigation and Monitoring Challenges in Software Organizations: A Morphological Analysis," *Int. J. Recent Innov. Trends Comput. Commun.*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:262209215

[59] S. Chaouch, A. Mejri, and S. A. Ghannouchi, "A framework for risk management in Scrum development process," *Procedia Comput. Sci.*, vol. 164, pp. 187–192, 2019.

[60] S. Beecham, T. Clear, R. Lal, and J. Noll, "Do scaling agile frameworks address global software development risks? An empirical study," *J. Syst. Softw.*, vol.

171, p. 110823, 2021.
[61] O. Erdoğan, M. E. Pekkaya, and H. Gök, "More effective sprint retrospective with statistical analysis," *J. Softw. Evol. Process*, vol. 30, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:48363402
[62] I. V Abramov, V. V Taratukhin, and I. V Illarionov, "A methodology for assessment and management of process-related risks," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:216079624
[63] O. E. Sandoval-Alfaro and R. R. Quintero-Meza, "Application of Data Analytics Techniques for Decision Making in the Retrospective Stage of the Agile Scrum Methodology," *2021 Mex. Int. Conf. Comput. Sci.*, pp. 1–8, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:237520139
[64] S. Zoltán and T. M. Tamás, "A Broader View of Risk Management Process in Projects," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226430251
[65] M. Cook and J. P. T. Mo, "Lifecycle Risk Modelling of Complex Projects," *Perspect. Risk, Assess. Manag. Paradig.*, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:117659161
[66] S. H. Ibrahim, A. Ali, and M. Sirshar, "Analysis of Software Project Complexity Factors of Large Scale Systems and Their Impacts on Core Knowledge Areas of Software Project Management," 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:208878766
[67] I. C. Satizábal-Echavarría and N. M. Acevedo-Quintana, "MePRiSIA: risk prevention methodology for academic information systems," *Rev. Fac. Ing. Univ. Antioquia*, no. 89, pp. 81–101, 2018.
[68] H. Haddad and F. Laghzaoui, "Review of risk management standards: Convergences and divergences," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229110133
[69] E. Kempe and A. Massey, "Regulatory and security standard compliance throughout the software development lifecycle," 2021.
[70] L. Y. Banowosari and B. A. Gifari, "System analysis and design using secure software development life cycle based on ISO 31000 and STRIDE. Case study mutiara ban workshop," in *2019 Fourth International Conference on Informatics and Computing (ICIC)*, IEEE, 2019, pp. 1–6.
[71] W. S. Wan Husin, Y. Yahya, N. F. Mohd Azmi, N. N. Amir Sjarif, S. Chuprat, and A. Azmi, "Risk Management Framework for Distributed Software Team: A Case Study of Telecommunication Company," *Procedia Comput. Sci.*, vol. 161, pp. 178–186, 2019, doi: https://doi.org/10.1016/j.procs.2019.11.113.
[72] S. K. Arora, "Project Failure: A Bad Communication (Case Study)," *Int. J. Manag. Humanit.*, 2023, [Online]. Available: https://api.semanticscholar.org/CorpusID:254843269
[73] M. Bourrier and C. Bieder, "Risk Communication for the Future," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:169261173
[74] A. Q. Mohabuth and B. N. Nankoo, "Towards the effectiveness of communication in adopting virtual team for software development," *Int. J. Eng. Comput. Sci.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:245224561
[75] J. Menezes Jr, C. Gusmão, and H. Moura, "Risk factors in software

development projects: a systematic literature review," *Softw. Qual. J.*, vol. 27, no. 3, pp. 1149–1174, Sep. 2019, doi: https://doi.org/10.1007/s11219-018-9427-5.

[76] S. M. Harris and L. Bronner, "Extension of the System Development Life Cycle (SDLC) for the Analysis of Complex Problems," *Int. J. Sci. Technol. Eng.*, vol. 5, pp. 102–107, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:116053458

[77] M. Ehrlich, G. Lukas, H. Trsek, J. Jasperneite, and C. Diedrich, "Investigation of Resource Constraints for the Automation of Industrial Security Risk Assessments," *2022 IEEE 18th Int. Conf. Fact. Commun. Syst.*, pp. 1–8, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:249101785

[78] S. Kumar, "Risk Culture Is a Necessary Condition for Enterprise Risk Management to Succeed," *ERN Uncertain. Risk Model.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:237562449

[79] R. M. Kachalov and Y. A. Sleptsova, "Organizational Culture in the Focus of Improving Risk Management in the Enterprise," *Issues Risk Anal.*, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:240853775

[80] T. Hussain, "Risk Management in Software Engineering: What Still Needs to Be Done," *Adv. Intell. Syst. Comput.*, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:169569338

[81] L. Sarana, O. Bilan, and I. Bitiuk, "A MANAGEMENT RISKS OF ENTERPRISE IS IN MODERN TERMS MENAGE," *Probl. Syst. APPROACH Econ.*, 2021, [Online]. Available: https://api.semanticscholar.org/CorpusID:242404525