Analysis of Risk Assessment Framework Using Agile Methodology and Computer-Aided Software Engineering Tools

Thapelo Monageng*1, Bukohwo Michael Esiefarienrhe2

1-2North-West University, Computer Science Department, South Africa

E-mail: 40177688@mynwu.ac.za1, Michael.Esiefarienrhe@nwu.ac.za2

Abstract. This research paper presents a risk assessment framework developed through the integration of the Scrum process with a Computer-Aided Software Engineering (CASE) tool, specifically Microsoft Excel. The framework aims to automate risk assessment processes, facilitating more accurate calculations of risk based on impact and probability using a matrix system. Initially, data is collected and recorded in the Risk Register, serving as the foundation for risk evaluation. The integration of the CASE tool enhances the efficiency and effectiveness of the process, allowing for real-time updates and tracking of risks. As updates occur within the Risk Register, a real-time dashboard is generated, providing stakeholders with an immediate overview of the risk landscape. The dashboard feature significantly improves decision making capabilities by presenting visual insights into risk levels and potential impacts on the projects. This dynamic monitoring tool is crucial for effective decision-making and timely responses to emerging risks. By leveraging agile methodologies design, this framework not only streamlines the risk management process but also ensures that relevant data is readily accessible to project teams. The findings from this research contribute to the fields of project management and software engineering by demonstrating the benefits of incorporating agile approaches with automated tools in risk assessment. The potential implications of this framework extend beyond individual projects, offering a model that can be adopted by various organizations seeking to enhance their risk management practices in a streamlined and efficient manner.

Keywords: Risk Assessment, Scrum, CASE Tool, Risk Register, Dashboard

1. Introduction

Scrum is a part of agile software development methods with the ability to respond to complex conditions and activities which ensures the success of the software development project [1]. Integrating agile methodologies with Computer-Aided Software Engineering (CASE) tools constitutes a significant progression in project management approaches [2]. Scrum, as one of the most widely adopted agile frameworks, has demonstrated considerable efficacy in addressing the inherent uncertainties and complexities of software development projects (Emmanuel et al., 2024). However, the systematic identification, analysis, and mitigation of risks within these agile contexts remains an area of research that requires further exploration, particularly when enhanced with technological innovations such as artificial intelligence (AI). This article examines the integration of Scrum methodologies with specialized CASE tools specifically focused on risk assessment frameworks and investigating the subsequent outcomes and implications for project success metrics.

The increasing complexity of software development projects, characterized by evolving requirements, technological uncertainties, and diverse stakeholder expectations, necessitates robust risk management practices. Traditional risk assessment approaches have often been criticized for their lack of adaptability and responsiveness to changing project conditions [4]. The emergence of agile methodologies like Scrum has introduced iterative and incremental approaches to development that inherently address certain risk factors through continuous feedback and adaptation [5] [6]. Nevertheless, the specific mechanisms by which Scrum practices interact with formalized risk assessment frameworks, particularly when mediated through CASE tools, warrant further investigation.

CASE tools have evolved considerably from their initial iterations as simple diagramming utilities to comprehensively supporting various aspects of the software development lifecycle. Though traditional CASE tools were perceived as a failure, recent innovations have enable them to be used for generating high-quality, defect-free, and maintainable software that are generally connected with methodologies for the development of information systems, as well as automated tools that can be utilized in a software development process [7] .

Contemporary CASE tools offer specialized functionalities for risk identification, analysis, monitoring, and mitigation, potentially complementing the iterative nature of Scrum processes. The intersection of these technological capabilities with Scrum's inherent risk management features presents a promising yet understudied domain within software engineering research. Several studies have examined either the risk management aspects of Scrum [8] or the application of CASE tools in software development contexts [9] [7]. However, limited research has specifically addressed the integration of these tools and the resulting impact on project outcomes. This research gap is particularly notable given the widespread adoption of both Scrum methodologies and various CASE tools in industry settings, creating a practical imperative for understanding their combined effects.

This article presents a comprehensive analysis of how Scrum practices, when integrated with risk assessment-focused CASE tools, influence project outcomes across multiple dimensions. The article addresses several key questions: How do the iterative cycles of Scrum interact with the structured risk assessment capabilities of CASE tools? What adaptations to standard Scrum ceremonies and deliverables emerge when supplemented by CASE tool risk data? How do these integrated approaches affect traditional project success metrics such as schedule adherence, budget compliance, quality outcomes, and stakeholder satisfaction? Additionally, the article examines whether certain categories of software development projects benefit more substantially from this integrated approach.

Through a mixed-methods research design incorporating both quantitative analysis of project performance metrics and qualitative assessment of practitioner experiences, this study provides insights into the practical implications of combining agile methodologies with specialized risk assessment tools. The findings contribute to both theoretical understanding of risk management in agile contexts and provide actionable guidance for practitioners seeking to enhance project outcomes through integrated approaches to develop methodology and technological support.

2. Literature Review

2.1. Agile Methodologies and Risk Management

The evolution of software development methodologies from traditional waterfall approaches to agile frameworks represents a paradigm shift on how projects are conceptualized, executed, and managed. Agile methodologies emerged as a response to the limitations of plan-driven approaches, particularly their inability to adapt to changing requirements and effectively manage risks in dynamic project environments [10] [11]. Among these methodologies, Scrum has gained significant popularity due to its structured yet flexible framework that accommodates iterative development and continuous feedback [5].

Risk management within agile contexts has been explored by numerous researchers, with varying perspectives on the inherent risk mitigation capabilities of agile practices. [12] argued that Scrum's iterative nature, with its short sprints and frequent deliverables, serves as an implicit risk management mechanism by allowing early identification of issues and rapid adaptation. This view is supported by [13], who found that many agile practitioners consider risk management to be embedded within agile practices rather than requiring separate formalized processes. Furthermore, [14] argued that agile methodologies usually focus on rapid development over hefty documentation which is contrary to the nature of the risk management process.

However, [15] [16] challenged this assumption, presenting evidence that implicit risk management in Scrum is insufficient for complex projects with significant uncertainty or regulatory requirements. Their research indicated that while Scrum provides mechanisms for handling certain types of risks, particularly those related to requirements and scope changes, it lacks structured approaches for identifying, assessing, and managing other risk categories such as technical, organizational, or external risks. This limitation highlights the potential value of integrating dedicated risk assessment frameworks with Scrum practices.

[17] presented a risk management framework for projects that use the agile methodology to help software development process and increase the likelihood of the project’s success while [18] presented a study to review project risk. Their findings reveals that while agile methods do address certain risks through their core practices, formal risk assessment and management are often overlooked in agile implementations. They identify a need for structured risk management that complements rather than contradicts agile principles, suggesting an opportunity for technological support through appropriate tools.

2.2. CASE Tools in Modern Software Development

CASE tools have evolved significantly from their origins as diagramming and documentation utilities. These tools support members of development team (including analysts, designers, coders, database administrators, and project managers) in building new software systems [7]. Contemporary CASE tools encompass a wide range of functionalities supporting various aspects of the software development lifecycle, including requirements management, design modelling, code generation, testing, and project management [19]. The adoption of these tools has been studied extensively, with research indicating both benefits and challenges in their implementation.

[20] examined the impact of CASE tools on software development productivity and quality, finding significant positive correlations between appropriate tool usage and improved project outcomes. Their results suggest that the quality of software developed using CASE tools are better than conventionally developed systems with respect to reliability, maintainability and portability but the degree of improvement is affected by the particular CASE tool used for development.

In the context of risk management, specialized CASE tools have emerged to support systematic risk assessment and mitigation activities. Other researchers surveyed risk management tool capabilities, identifying key functionalities including risk identification catalogues, probability and impact assessment matrices, mitigation planning features, and monitoring dashboards [21]; [22]; [23]; [24]. Their research suggests that these tools can enhance risk visibility and facilitate more consistent risk management practices across project teams.

However, [25] cautioned that the introduction of specialized tools can create friction when they do not align with established development workflows. Their case studies demonstrated that tools perceived as burdensome or disconnected from daily development activities often face adoption challenges, regardless of their potential benefits. This highlights the importance of examining how risk-focused CASE tools integrate with Scrum practices and team dynamics.

2.3. Integration of Scrum and CASE Tools
The intersection of Scrum methodologies and CASE tools represents an evolving area of research and practice. [26] investigated the use of development tools in large-scale agile projects, concluding that prominent challenges are a lack of testing strategies, chaos in sprint execution and deadlines, ignoring coding standards, and requirement scoping. [27] presents some relevant agile tools that could improve every software development project and found that tool selection and integration significantly influence the effectiveness of agile implementations. These research indicates that successful tool integration requires alignment with agile principles and team workflows, suggesting that risk assessment tools must be compatible with Scrum's iterative and collaborative nature to be effective.

[28] proposed a comprehensive framework aimed at integrating risk management into agile development projects, specifically within the Scrum methodology, by leveraging guidance from the Project Management Body of Knowledge (PMBOK). The primary objectives of this framework are to enhance the risk management processes within Scrum and to improve the overall success rate of Scrum projects. This approach bridges traditional risk management practices with agile principles, addressing the need for systematic risk handling in fast-paced, iterative environments. Their work contributed to literature by offering a structured method to embed risk management seamlessly into Scrum, thus supporting better project outcomes and increased project success rates.

2.4. Risk Assessment Frameworks in Software Development
Risk assessment frameworks provide structured approaches to identifying, analysing, and managing project risks. Several frameworks have been developed specifically for software development contexts, including Boehm's risk management framework [29], the Software Engineering Institute's risk management paradigm [30], and the Riskit method [31]. These frameworks typically involve processes for risk identification, analysis, prioritization, mitigation planning, and monitoring.

More recent research has explored adaptations of these frameworks for agile contexts. [32] proposed a lightweight risk management framework designed to complement Scrum practices, emphasizing integration with sprint planning and retrospective activities. Their empirical evaluation suggests that such adapted frameworks can enhance project outcomes without compromising agile principles.

[33] proposed a new approach to perform risk assessment and management of agile projects comprising a set of qualitative risk tools in-line with the flexibility found in agile methods that introduces a systematic way of risk analysis in agile projects.

2.5. Empirical Studies on Outcomes and Performance Metrics
Empirical research on the outcomes of integrating risk assessment frameworks with Scrum methodologies remains limited, with most studies focusing on either risk management in agile contexts or tool adoption separately. [34] explores the adoption of Behavior-Driven Development (BDD) practices within Scrum teams and proposes a metrics framework designed to enhance Scrum processes and product quality. The findings highlight that BDD significantly contributes to improving collaboration, communication, and the understanding of requirements among Scrum team members, thereby supporting more effective and quality-driven project outcomes.

[35] investigated the human-related critical success factors for the success of agile software development projects. The study identifies team capability and customer involvement as the most influential factors, with psychological safety playing a significant indirect role. Additionally, team autonomy exhibits a complex, competing relationship with psychological safety, highlighting the nuanced interplay of human factors in agile project success.

The research by [36] investigated the success factors of agile software development in developing countries, with a focus on Bangladesh. Addressing a research gap, it employs mixed methods surveys, observations, and interviews to identify key components influencing project success. The study highlights the critical importance of active customer engagement and cultural considerations, underscoring the need to tailor agile practices to local contexts. Findings suggest that understanding these factors can improve agile implementation and reduce project failures in Bangladesh, offering valuable insights for software enterprises operating in similar environments.

2.6. Research Gap and Contribution

The literature review reveals several significant gaps in current understanding of how Scrum methodologies integrate with CASE tool-supported risk assessment frameworks:

1. Limited empirical research specifically examining the integration of Scrum practices with risk assessment CASE tools.
2. Insufficient exploration of how risk data from CASE tools can be effectively incorporated into Scrum artifacts and ceremonies.
3. A lack of comprehensive studies measuring the impact of this integration on multiple project outcome dimensions.
4. Minimal investigation into contextual factors that might influence the effectiveness of integrated approaches across different project types or organizational environments.

This research aims to address these gaps by providing empirical evidence on the implementation and outcomes of Scrum methodologies enhanced with CASE tool-supported risk assessment frameworks. By examining both process adaptations and resulting project metrics, the study contributes to both theoretical understanding and practical guidance for software development teams seeking to enhance their risk management capabilities while maintaining agile principles.

3. Methodology

The Electronic Risk Assessment Framework (E-RAF) was developed and implemented using a systematic process that combined agile software development principles, notably Scrum, with practical tool creation in Microsoft Excel. This section outlines methodology used to design, develop, and evaluate the Electronic Risk Assessment Framework (E-RAF), the use of Scrum in the development lifecycle, the technical implementation in Excel, and the process for data collection and analysis to validate the risk assessment framework.

In this study, the mixed-methods design combined quantitative and qualitative approaches to provide a fuller picture of the framework’s impact. On the quantitative side, project performance metrics such as schedule adherence, cost control, and risk reduction trends were generated and analysed through the Excel-based E-RAF tool using probability–impact matrices and automated dashboards. Complementing this, the qualitative side drew on practitioner experiences gathered during Scrum ceremonies, where feedback on tool usability, decision-making, and team interactions was documented. Together, these strands allowed us to capture not only the measurable outcomes of the framework but also the lived experiences of those applying it in practice.

The methodology combines agile software development, specifically the Scrum framework, with lightweight tool development in Microsoft Excel to produce a cost-effective and easily deployed risk management solution. The reason for merging Scrum with a spreadsheet-based CASE (Computer-Aided Software Engineering) tool is in two-fold: to improve responsiveness to changing user needs through iterative development and to take advantage of Excel's ubiquity and flexibility for rapid prototyping and implementation.

Scrum is a widely adopted agile methodology that emphasizes iterative progress, cross-functional collaboration, and adaptive planning. It is particularly well suited to software projects that require strong stakeholder interaction and frequent input, which is especially important for designing decision-support tools such as E-RAF. The CASE tool portion of this project is consistent with the trend of using low-code/no-code platforms to simplify software development, lower costs, and shorten time-to-delivery.

Excel was chosen as the platform for tool development due to its widespread availability in organizational environments, built-in support for data processing and visualization, and the ability to extend its functionality using Visual Basic for Applications (VBA). Furthermore, the tool's foundation is based on risk assessment approaches, which promote an organized approach to risk identification, analysis, and treatment.

The methodology is presented in four parts: (1) the conceptual design of the E-RAF framework, (2) the Scrum-based software development lifecycle, (3) the technical implementation of E-RAF in Excel, and (4) the data collection and analysis process used to validate and evaluate the framework's effectiveness.

3.1. Framework Design

The E-RAF framework was designed to support systematic identification, analysis, evaluation, and monitoring of project risks. The design was informed by established risk management standards and customized to be lightweight, user-friendly, and compatible with common project management practices. The key components of the E-RAF are:

- Risk Identification Module: Enables users to enter and categorize risks using predefined categories such as financial, technical, operational, and external.
- Risk Scoring Engine: Calculates risk scores using a probability-impact matrix with customizable scales.
- Risk Prioritization**:** Automatically ranks risks based on their composite score and flags high-priority items.
- Reporting Interface: Generates visual outputs, such as risk heat-maps and summary tables, for decision-making.

A data flow model and modular design approach were adopted to ensure scalability and maintainability. The architecture was intentionally lightweight to accommodate deployment within Excel, requiring no external database or dependencies.

3.2. Scrum Implementation

The Scrum methodology was selected to enhance adaptability, foster team collaboration, and ensure iterative refinement of the E-RAF tool. The development team comprised a Product Owner, Scrum Master, Accountant, Security specialist and Operations Manager and three Developers. The Scrum process was implemented as follows:

Product Backlog Creation: Functional and non-functional requirements were converted into user stories and meticulously documented in the product backlog. To help with sprint planning, each story provided acceptance criteria, a priority level, and an anticipated effort. This organized backlog allowed the Scrum team to deal with risk tool features incrementally, taking into account user value and technical feasibility.

Sprint Planning: The project was divided into 5 sprints, each lasting two weeks. Sprint planning meetings were held to select user stories from the backlog and define the sprint goals. Daily Scrum short stand-up meetings were conducted to assess progress, identify blockers, and coordinate tasks.

Sprint Review: At the end of each sprint, a working increment of E-RAF was demonstrated to stakeholders for feedback.

Sprint Retrospective: Team members reflected on the sprint process to identify areas for improvement and update workflow practices accordingly. Scrum artefacts including Product Backlog, Sprint Backlog, and Burndown Charts were maintained using Excel sheets to ensure transparency and traceability.

3.3. Tool Development in Excel

Excel was chosen as the development platform because it is widely used, easy to install, and has built-in data handling features. The E-RAF application was created by combining Excel capabilities and VBA scripting to emulate CASE environment. The E-RAF in Excel was created to meet the government need for an accessible, low-cost, and simply deployable risk management solution. While traditional risk assessment tools frequently require specialized software platforms or extensive training, Excel was chosen due to its familiarity among project managers, powerful built-in functionalities, and flexibility to support data manipulation and visualization tasks. Development features included:

3.3.1. Data Input Forms

Structured input forms were created using Excel's built-in form capabilities, such as drop-down lists, choice buttons, and date pickers, to streamline risk data entry. These controls guaranteed that data was collected consistently, eliminated input errors, and directed users through necessary fields. The forms improved user experience and data integrity throughout the application by arranging inputs into logical areas (for example, risk descriptions, probability/impact scoring, and ownership).

3.3.2. Visualizations

Conditional formatting was used across the risk register to constantly emphasize risk levels based on their calculated severity (for example, red for "Very High" and green for "Very Low"). This enabled users to receive rapid visual feedback when entering or evaluating data. Furthermore, Excel chart objects were used to create real-time dashboards with pie charts and heat maps that summarized risk distribution by category, status, and severity.

3.3.3. Automated Calculations

The ability to prioritize risks using a systematic and quantitative way is a key feature of E-RAF. The 5×5 Probability-Impact Matrix is commonly used in project management and risk analysis for its simplicity and efficacy. The E-RAF tool evaluates each identified risk in two key dimensions: probability (P) and impact (I). Each dimension is assigned a score on a five-point scale, with 1 being the lowest and 5 being the greatest. A Probability score of 1 may denote a rare occurrence, but a score of 5 suggests that the risk is extremely probable to materialize (Table 1). Similarly, an Impact score of 1 may imply insignificant ramifications, whereas a 5 indicates catastrophic effects on the project's scope, schedule, cost, or quality.
Once both scores are assigned, the **Risk Rating** (RR) is computed as shown in eqn 1.

$$\mathbf{RR = P \times I} \tag{1}$$

This RR generates a composite score ranging from 1 to 25, which is then translated into qualitative risk levels using a predefined classification grid as shown in table 1incorporated in the matrix.

Table 1. Risk Rating Table

RR	Risk Level	Colour Code
20–25	Very High	Red
15–19	High	Orange
10–14	Moderate	Yellow
5–9	Low	Light Green

3.3.4. Objective Prioritization

Quantifying risks reduces subjective bias when determining criticality. A risk with high probability but low impact is treated differently from one with low probability but potentially severe consequences. This structured approach enables project teams to prioritize and focus their mitigation efforts on the risks that present the most significant threat to project success.

E-RAF converts the 5×5 matrix into an interactive risk heatmap that displays each risk's likelihood and impact scores. The matrix cells are color-coded to indicate risk intensity, providing users with a quick overview of the risk environment. High and Very High risks (red and orange zones) are clearly identifiable from Moderate or Low risks, allowing for quick, judgments. The modular spreadsheet structure separated raw data, computation logic, and output visualization, thereby following good practices of software modularity and separation of concerns.

3.4. Data Collection and Risk Analysis

To evaluate the performance and validity of the E-RAF tool, a custom dataset was developed using a combination of anonymized historical records and simulated project scenarios. This approach was necessary due to the absence of a publicly available, domain-specific dataset reflecting the dynamic nature of Agile risk environments in the public sector.

The dataset used in this study comprised two primary sources:

- Anonymized risk logs sourced from a mixture of archived ICT and construction project documents previously managed by consulting firms and government departments. These records were stripped of identifying details and used to model realistic project risks and mitigation workflows.
- Simulated risk scenarios were created to reflect a range of sprint-level risks commonly encountered in Agile e-government project environments. These scenarios were designed to include variation in risk type, probability, and impact, enabling comprehensive testing of the E-RAF framework across multiple project conditions.

Due to the hybrid nature of the dataset and the confidentiality agreements associated with the original materials, the full dataset is not publicly hosted. However, key samples and anonymized structures used in the evaluation can be provided upon request for validation or replication purposes. This method aligns with established research protocols where **synthetic** or **non-publicly available data** is utilized to validate tools under realistic yet controlled conditions [37] [38].

3.5. Parameters

3.5.1. Registration Parameters

Registration parameters shown in table 2 capture the initial details of each risk, including its description, probability, impact, owner, and calculated risk level. These inputs form the foundation for prioritization and monitoring risks throughout the project lifecycle.

Table 2. Registration parameters

Parameter	Description
Risk Id	Unique identifier for each risk (e.g., R-001)
Risk Description	Brief description of the identified risk
Sprint No	Associated sprint number in Scrum cycle
Risk Owner	Person responsible for managing the risk
Status	Current status of the risk (e.g., Open, Closed)
Email	Contact email of the risk owner
Registration Date	Date the risk was registered
Completion Date	Planned or actual risk resolution date
Probability (P)	Likelihood of risk occurrence (scale: 1–5)
Impact (I)	Severity of impact if the risk occurs (scale: 1–5)
Rating (P × I)	Calculated risk score based on probability and impact
Risk Level	Categorized level of risk (e.g., Very High, High, Moderate, Low, Very Low)

3.5.2. Mitigation Parameters

Mitigation parameters shown in table 3 documents the planned response to each risk, including treatment actions, responsible persons, status, and overdue days. They help track the effectiveness and timeliness of mitigation efforts within each sprint.

Table 3. Mitigation parameters

Parameter	Description
Risk Id	Cross-reference to the Registration_Form
Risk Description	Auto-filled from Registration_Form

Parameter	Description
Sprint No	Auto-filled from RegistrationForm
Treatment Action	Description of the planned mitigation or response
Responsible Person	Auto-filled from Registration_Form (Risk Owner)
Date Registered	Auto-filled from Registration_Form
Completion Date	Expected or actual date the mitigation is complete
Days Overdue	Calculated as: TODAY() − Completion Date (if Status is Open and past due)
Status	Current status of the mitigation plan (Open or Closed)

Table 4. Registration_Form

Risk Id	Risk Description	Sprint No	Risk Owner	Status	Email	Registration Date	Completion Date	Probability (P)	Impact (I)	Rating (P×I)	Risk Level
RSK-001	Delay in component delivery	Sprint 2	admin	Open	admin@example.com	2025-04-10	2025-05-05	4	5	20	Very High

The **Registration_Form** is a structured interface within the E-RAF tool used to capture and store initial risk details during Agile sprint cycles. It serves as the primary entry point for documenting risks at the point of identification and plays a critical role in standardizing how risk information is collected across stakeholders.

The form ensures consistency by using dropdowns, score ranges, and date validation. When a user submits a new risk, the form automatically calculates the **rating** by multiplying the **probability (P)** and **impact (I)**. The result is then categorized into a **risk level**, using a predefined matrix.

Calculation Notes:

- Rating = Probability × Impact = 4 × 5 = 20
- Risk Level Thresholds (customizable):
    - 20–25: Very High
    - 15–19: High
    - 10–14: Moderate
    - 5–9: Low
    - 1–4: Very Low

The use of the Registration_Form standardizes the risk entry process and enables accurate comparison and prioritization across different risks. All submissions feed directly into the hidden Risk_Register sheet, where they are stored for tracking, reporting, and evaluation throughout the project lifecycle.

Table 5. Mitigation

Risk Id	Risk Description	Sprint No	Treatment Action	Responsible Person	Date Registered	Completion Date	Days Overdue	Status
RSK-001	Delay in component delivery	Sprint 2	Expedite order through alternate supplier	Scrum Master	2025-04-10	2025-05-05	5	Open

Table 5 presents a structured summary of risk mitigation activities recorded during the implementation of the E-RAF framework. Each row represents a specific risk entry, linking it to the corresponding **mitigation action, responsible person**, and relevant timeline information. The table captures key operational details such as the **date the risk was registered,** the **planned completion date**, and the calculated number of **days overdue**, which reflects the timeliness of response efforts.

In the example shown, Risk ID **RSK-001** concerns a delay in component delivery identified in **Sprint 2.** The mitigation strategy involved expediting the order through an alternate supplier, with **Scrum Master** assigned as the responsible party. The risk remained **open** as of the recorded date, with a **5-day delay** beyond the intended resolution date. This table supports accountability and tracking by clearly linking risks to assigned mitigation actions and response performance, thereby enhancing transparency and improving decision-making within Agile project cycles.

The "Days Overdue" column is calculated by subtracting the planned **Completion Date** from the current date, highlighting delays in mitigation response. This metric supports real-time performance monitoring and feeds into dashboard visuals that track unresolved or overdue risks. Calculations below shows how to calculate Days overdue.

Calculation Notes:

- Days Overdue = IF(Status = Open AND TODAY() > Completion Date, TODAY() - Completion Date, 0)
    - For example, (assuming today is 2025-05-10): = 2025-05-10 – 2025-05-05 = 5
- Conditional formatting was used to highlight overdue tasks when Days Overdue > 0.

4. Implementation

Risk assessment plays a central role in the success and sustainability of project management, particularly in high-complexity domains such as IT, infrastructure, and e-government projects. Effective identification, analysis, and mitigation of risks ensure that project objectives are met within scope, budget, and timeline constraints [39]. Despite widespread awareness, many projects continue to fail due to inadequate risk monitoring tools and practices, even when teams are aware of the risks. Recent research shows that weak monitoring and control processes are significantly linked to lower project success rates [40].

Traditional risk assessment methods such as qualitative risk matrices and static risk logs often fall short in supporting continuous and adaptive project environments. These approaches are not seamlessly integrated into iterative development workflows, lack real-time responsiveness, and typically rely on rigid structures with limited scalability [41]; [42]. These limitations make them unsuitable for dynamic environments where risk factors evolve rapidly, and decisions must be made continuously based on updated information.

To address these gaps, this study introduces the **Excel-based Risk Assessment Framework (E-RAF)**, a lightweight, agile-compatible CASE tool that combines the familiarity and accessibility of Excel with the adaptability of Scrum-based iterative development. Excel was chosen not merely for its simplicity but for its powerful formula engine, data visualization capabilities, and ability to be customized rapidly without extensive programming overhead.

The integration of Scrum methodology serves to introduce agility into the tool development process and the risk assessment workflow itself. Through iterative **software-use phases**, E-RAF evolves with ongoing stakeholder feedback, delivering incremental improvements in usability, automation, and analytical accuracy.

This paper contributes to research and practice

- Presenting a novel integration of **Scrum and CASE tool development** within an Excel environment for risk management.
- Demonstrating how **risk matrices, heatmaps, and mitigation tracking** can be automated and visualized using Excel's native capabilities.
- Validating the E-RAF tool through **simulated and real project data,** showcasing effectiveness in iterative risk control.
- Comparing E-RAF with traditional tools such as **Primavera Risk Analysis** and generic Excel logs to highlight its relative advantages.

By leveraging Agile values and low-code technologies, E-RAF offers an accessible, iterative, and visually driven solution that enables organizations strengthen their risk management practices without the need for complex software investments.

4.1. Planning and Execution

The development of the E-RAF tool followed a structured phase development to ensure accuracy and alignment to the Case tool development requirements. The case tool was developed in phases.

4.2. Phase Planning and Execution

The development of the E-RAF tool was organized into five iterative phases; each designed to incrementally deliver functional improvements and ensure alignment with agile principles and stakeholder expectations. These phases facilitated a structured yet flexible approach to CASE tool development, with each phase incorporating planning, implementation, testing, and review cycles.

Phase 1 entails requirement gathering and core architecture design. The risk register's initial structure as well as interactive data entry forms that use Excel's built-in form controls were developed. It allowed users to enter critical risk data such as descriptions, classifications, probability, and impact, which would then be used to conduct further analyses.

Phase 2 introduced the risk evaluation engine matrix format. Risk ratings were automatically calculated by multiplying the assigned probability and impact scores, with results classified into five severity levels (Very Low to Very High). Conditional formatting was applied to color-code risk cells, enhancing the visibility of critical risk areas for prioritization.

Phase 3 enhanced the tool's capabilities by incorporating mitigation planning component. This phase used lookup methods to tie mitigation measures directly to registered risks, as well as introduced calculated values such as status, completion date, and days overdue. These features enabled real-time tracking of mitigation measures and closure deadlines.

Phase 4 focused on data visualization and performance tracking. Dashboards were created using pivot tables and dynamic chart objects, offering graphical representations of risk distributions, closure trends, and risk category breakdowns. Heatmaps across software-use phases and visual trendlines provided stakeholders with clear insights into evolving risk profiles.

Phase 5 emphasized refinement and usability enhancements. Macro-driven menus were introduced to streamline navigation, automate report generation, and simulate version control.

The E-RAF tool generated a comprehensive set of risk assessment results, providing both quantitative and qualitative insights into project vulnerabilities throughout and agile sprints. For each sprint cycle, the program created a risk register with unique risk identifiers, complete descriptions, associated sprint numbers, risk owners, probability and impact scores, and automatically calculated severity ratings. These ratings were converted into category risk levels (Very Low to Very High) using a 5×5 scoring matrix. The system also monitored real-time status changes, mitigation efforts, outstanding tasks, and closure rates.

These outputs directly correspond to the Scrum stages that structured the project workflow. During **Sprint Planning,** the probability–impact matrix and risk categorization rules were agreed upon, which are reflected in the standardized registration parameters (Table 2) and automated scoring matrices (Tables 6 and 7). In the **Daily Scrum** meetings, newly identified risks and mitigation updates were logged in real time, feeding into the risk register (Table 2) and mitigation tracking sheets (Table 3 and Table 5). This continuous updating ensured that the dashboards and heatmaps represented the latest sprint conditions. At the end of each sprint, during the **Sprint Review,** stakeholders evaluated the dashboards and heatmaps, such as the observed decline in "Very High" risks between Sprints 2–5 (Table 8). Finally, the **Sprint Retrospectives** enabled the team to refine mitigation approaches and visualization features, visible in the improvements to overdue tracking (Table 5) and dynamic dashboards (Table 9) introduced in later sprints.

Visual heat maps made it easy to identify high-risk zones across sprints, while trend charts showed how risk intensity and mitigation efficacy changed over time. The summary dashboard also included indicators such as total open and closed risks, overdue items, and category-specific distributions (technical, financial, security, and operational), providing a comprehensive perspective of project risk posture. These outputs not only offered situational knowledge to project teams but also acted as a strategic tool for sprint planning, resource allocation, and stakeholder reporting.

4.3. Model Building Calculations and Results

The core analytical output of the E-RAF tool is a **5×5 risk matrix,** automatically generated based on the assigned Probability (P) and Impact (I) scores for each identified risk. This matrix serves as a central reference for visualizing the severity and priority of risks within the project lifecycle. Each cell in the matrix represents a calculated **Risk Rating** (P × I), with values ranging from **1 (Very Low)** to **25 (Very High).** Risks are categorized into five color-coded levels: **Very Low, Low, Medium, High,** and **Very High**, providing an immediate visual aid for decision-making as depicted in Table 6.

Table 6. Risk rating matrices

IMPACT	1	2	3	4	5
5	1	2	3	4	5
4	2	4	6	8	10
3	3	6	9	12	15
2	4	8	12	16	20
1	5	10	15	20	25

Each risk item in the registration form is scored using a Risk Rating formula:

- Risk Rating=Probability (P)×Impact (I)

Probability and Impact are each rated on a scale of 1 (Very Low) to 5 (Very High).
The resulting score (ranging from 1 to 25) determines the Risk Level using a predefined 5×5 matrix shown in table 7 and each new risk entered is automatically classified, and counts are tallied per sprint, feeding directly into the heat map.

Table 7. Risk Calculation Formula

Rating (P×I)	Risk Level
1–5	Very Low
6–10	Low
11–15	Medium
16–20	High
21–25	Very High

```
=IF(L2>=21,"Very High",
 IF(L2>=16,"High",
 IF(L2>=11,"Medium",
 IF(L2<=6,"Low","Very Low"))))
```

Formula

In a **summary table** (the matrix that feeds the heat map), use the **COUNTIFS** function to tally risks for each combination of Sprint Number and Risk Level. Example formula:

```
= COUNTIFS(Registration_Form!$SprintNo, "Sprint 1",
        Registration_Form!$Risk_Level, "Very High")
```

The heatmap of risk trends across five sprints, showing how risks of various severity levels evolved Table 8. By interpreting the above feed, it can be concluded that:

Table 8. Heatmap

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
Very High	4	5	0	0	0
High	1	2	3	2	2
Medium	1	0	5	6	0
Low	0	0	0	0	0
Very Low	0	0	0	0	6

- **Sprint 1 and 2:** These contain the highest number of *Very High* risks (4 and 5 respectively), indicating elevated project vulnerability. The heat map cells for these values appear in **dark red**, prompting urgent risk mitigation.
- **Sprint 3 and 4:** We observe a shift, with *Very High* risks disappearing with an increase in *Medium* and *High* risks. This suggests that while critical threats were addressed, new risks continued to emerge showing characteristic of dynamic project environments.
- **Sprint 5:** Shows a risk reduction trend with no High or Medium risks and six *Very Low* risks. This transition toward green in the heat map represents improved risk handling, system stability, and maturity in mitigation planning.

The table below (Table 9) presents key metrics used to monitor and evaluate risks within the E-RAF framework. Each metric is calculated using automated Excel formulas to support real-time tracking and reporting.

Table 9. Summary of Automated Risk Monitoring Metrics and Formulas

Summary Metric	Formula
Total Open Risks	=COUNTA(RiskRegister!A4:A100)
Total No of Closed	=COUNTIF(RiskAssessment_PlanTracking!I4; "CLOSED")
No of Overdue	=COUNTIF(RiskAssessment_PlanTracking!I4; "CLOSED")
Risk by Category Technical	=COUNTIFS(RiskAssessment_PlanTracking!H:H; "OVERDUE")
Risk by Category (Security)	=COUNTIF(RiskRegister!B:B; "Technical")
Risk by Category (Operational)	=COUNTIF(RiskRegister!B:B; "Security")
Risk by Category (Financial)	=COUNTIF(RiskRegister!B:B; "Operational")

These formulas are used to derive **key risk indicators** in real-time from the underlying data sheets (e.g., RiskRegister, RiskAssessment_PlanTracking). Each metric is a foundational element for constructing the **project risk dashboard**. The calculated values from these formulas are displayed in a **Dashboard sheet** within Excel. The dashboard includes:

- KPI showing real time counts of Open, Closed, and Overdue risks.
- Pie Charts or Bar Graphs breaking down risk types (Technical), Security, Operational, and Financial.
- Conditional formatting to highlight high overdue risk levels
- Trends charts showing progress over sprints using these counts.

By linking these formulas **to dynamic named ranges** and **data visualizations**, the E-RAF tool delivers an up-to-date and interactive project risk dashboard. This enhances project governance, executive reporting, and sprint planning in e-government and other complex project environments.

Comparison with near similar tools

The comparison of E-RAF, Primavera Risk Analysis, and generic Excel (Table 10)-based risk logs reveals various trade-offs in cost, usability, scalability, and methodological depth. Primavera Risk Analysis, while robust and feature-rich, is generally intended for large-scale infrastructure projects with professional risk analysts, which frequently necessitate Monte Carlo simulations and complex probabilistic modelling. Its high cost and learning curve make it less accessible to small teams or public sector initiatives with limited funding. Generic Excel logs, on the other hand, are the most basic type of risk documentation, with no structure, analytical depth, or integration with iterative project frameworks.

They frequently suffer from inconsistency, manual errors, and insufficient insight generation. E-RAF, which sits between these two extremes, demonstrates how a well-structured Excel-based CASE tool can contain complex functionality such as automated scoring, conditional heat maps, sprint-wise tracking, and mitigation dashboards without requiring a large software investment. It uses basic spreadsheet capabilities, reinforced by VBA, to provide iterative risk monitoring in Agile-driven environments, particularly for digital government initiatives that require transparency and adaptability.

Table 10. Comparison Table

Feature	E-RAF (Excel-based)	Primavera Risk Analysis	Generic Excel Logs
Platform Cost	Low (Free with Excel)	High (Commercial License)	Low
User Accessibility	High (Familiar Interface)	Medium	High
Risk Scoring Automation	Yes	Yes	No
Visual Dashboards	Yes (Heatmaps, Charts)	Yes	No
Collaboration Support	Limited (File Sharing)	High (Team Integrations)	Low
Training Requirement	Low	High	Low
Customization Flexibility	High	Medium	Medium
Advanced Analytics (e.g., Monte Carlo)	No	Yes	No
Compliance with ISO 31000	Yes	Yes	Partially

5. Discussion

The use of the Excel-based Electronic Risk Assessment Framework (E-RAF) in a Scrum environment reveals some critical insights into risk management in agile projects. The findings show that incorporating structured risk assessment methods into iterative Scrum cycles can result in substantial gains in risk visibility, prioritization, and mitigation efficacy. The observed reduction in "Very High" risks over sprints demonstrates how iterative monitoring and ongoing feedback allow teams to respond proactively to emergent difficulties rather than relying on static risk registers.

This finding aligns with previous studies that argue agile practices implicitly address certain risks through frequent feedback and adaptation [12][13]. However, unlike purely implicit approaches, the E-RAF framework provides a structured and quantitative mechanism that complements Scrum practices. By embedding probability–impact matrices, dashboards, and overdue tracking into the sprint workflow, the framework addresses critiques that Scrum alone lacks sufficient mechanisms for managing complex or non-requirements-related risks [15] [16].

Another key outcome is the demonstration that Microsoft Excel, often overlooked as a CASE tool, can serve as an effective, low-cost platform for implementing risk management frameworks. While enterprise solutions such as Primavera Risk Analysis provide advanced capabilities, the results show that even lightweight spreadsheet-based systems can automate scoring, generate visual heatmaps, and support decision-making in agile environments. This bridges the gap identified in prior literature between expensive enterprise tools and overly simplistic manual logs [25] [26].

The integration of Scrum ceremonies with the E-RAF results further illustrates the value of combining process and technology. Sprint Planning facilitated the adoption of standardized scoring methods; Daily Scrums ensured risks were logged and tracked in real time; Sprint Reviews enabled stakeholders to evaluate dashboards; and Retrospectives contributed to iterative tool refinements. This process-driven visibility supports the argument that successful risk management in agile settings requires alignment between tools and team practices [26] [28].

Despite these contributions, some limitations should be acknowledged. The evaluation relied partly on simulated datasets, which may not capture the full complexity of real-world project environments. Additionally, Excel's file-based architecture limits its scalability for large, distributed teams. These constraints suggest that while the E-RAF is effective for small- to medium-scale projects, larger organizations may require web-based or integrated platforms.

Overall, the discussion emphasizes that systematic, tool-based risk assessment can coexist with Scrum's flexibility. The study contributes to the current literature by indicating that agile risk management benefits from hybrid techniques that combine quantitative monitoring with qualitative practitioner feedback, resulting in a more comprehensive awareness of project risks and mitigation.

6. Conclusion and future work

This study presented the development and evaluation of the E-RAF (Excel-based Risk Assessment Framework), a CASE tool designed to support risk management in Agile project environments, particularly for e-government implementations. The findings reveal important insights into how iterative Scrum cycles interact with structured risk assessment processes, and how this integration influences project outcomes. The integration of Scrum methodology with Excel's extensive automation and visualization features enabled the organized capture, classification, prioritizing, and mitigation of risks throughout five sprint cycles. The matrix-based risk scoring system, dynamic dashboards, and automated heat maps demonstrated that meaningful risk analysis can be achieved even with limited resources, leveraging widely accessible platforms like Excel.

The iterative cycles of Scrum provided the feedback loops necessary for continuous risk identification, monitoring, and mitigation. By embedding probability impact matrices and dashboards into sprint workflows, the framework transformed risk management from a static, document-based activity into a dynamic process aligned with Scrum's incremental delivery model. Further, the integration of CASE tool risk data prompted adaptations to standard Scrum ceremonies and deliverables. Sprint Planning incorporated structured risk scoring as part of backlog prioritization; Daily Scrums became points for updating risk registers in real time; Sprint Reviews leveraged dashboards to evaluate progress against risks; and Sprint.

The research contributes to both academic inquiry and practical application by showcasing how CASE tool development can benefit from agile principles and how Excel can be repurposed as a lightweight but effective development environment for risk management. It bridges the gap between high-cost enterprise risk tools and over simplified generic logs by offering a middle-ground solution that is affordable, customizable, and iterative.

However, the current implementation has limitations. Being built in Excel, E-RAF is constrained by file-based operations, limited scalability which may restrict its utility in large-scale, collaborative environments. Additionally, while VBA scripts and conditional formatting provide a level of automation, they are dependent on the user's familiarity with Excel's advanced features, which could affect adoption.

Future work will focus on extending E-RAF into a web-based application with real-time multi-user access, improved security, and broader integration with popular project management tools MS Project. These enhancements aim to scale the tool for enterprise use while preserving its core strengths simplicity, accessibility, and alignment with agile principles. With further refinement, E-RAF has the potential to evolve into a full-fledged, cross-platform risk management solution tailored for iterative and transparent project delivery models.

7. References

[1] E. P. Wonohardjo, R. F. Sunaryo, and Y. Sudiyono, ‘A Systematic Review of SCRUM in Software Development’, *JOIV : Int. J. Inform. Visualization*, vol. 3, no. 2, pp. 108–112, Mar. 2019, doi: 10.30630/joiv.3.2.167.

[2] Thapelo Monageng, ‘Development using Scrum: A CASE Tool Approach’, vol. 13, no. 2, pp. 1542–1549, 2025.

[3] Emmanuel Chibuike Daraojimba, Chinedu Nnamdi Nwasike, Abimbola Oluwatoyin Adegbite, Chinedu Alex Ezeigweneme, and Joachim Osheyor Gidiagba, ‘COMPREHENSIVE REVIEW OF AGILE METHODOLOGIES IN PROJECT MANAGEMENT’, *Comput. sci. IT res. j.*, vol. 5, no. 1, pp. 190–218, Jan. 2024, doi: 10.51594/csitrj.v5i1.717.

[4] Q. Meng, X. Qu, K. T. Yong, and Y. H. Wong, ‘QRA Model-Based Risk Impact Analysis of Traffic Flow in Urban Road Tunnels’, *Risk Analysis*, vol. 31, no. 12, pp. 1872–1882, Dec. 2011, doi: 10.1111/j.1539-6924.2011.01624.x.

[5] K. Schwaber and J. Sutherland, *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. 2020. [Online]. Available: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100

[6] B. M. Esiefarienrhe and T. Monageng, ‘Critical success factors frameworks, and models for risk assessment of eGovernment projects: A systematic literature review’, *J. Infras. Policy. Dev.*, vol. 8, no. 8, p. 4503, Aug. 2024, doi: 10.24294/jipd.v8i8.4503.

[7] S. Shafiee, Y. Wautelet, S. C. Friis, L. Lis, U. Harlou, and L. Hvam, ‘Evaluating the benefits of a computer-aided software engineering tool to develop and document product configuration systems’, *Computers in Industry*, vol. 128, p. 103432, Jun. 2021, doi: 10.1016/j.compind.2021.103432.

[8] S. Lopes, R. Gratão De Souza, A. Contessoto, A. Luiz De Oliveira, and R. Braga, ‘A Risk Management Framework for Scrum Projects’:, in *Proceedings of the 23rd International Conference on Enterprise Information Systems*, Online Streaming, --- Select a Country ---: SCITEPRESS - Science and Technology Publications, 2021, pp. 30–40. doi: 10.5220/0010448300300040.

[9] M. Limayem, M. Khalifa, and W. W. Chin, ‘CASE Tools Usage and Impact on System Development Performance’, *Journal of Organizational Computing and Electronic Commerce*, vol. 14, no. 3, pp. 153–174, Sep. 2004, doi: 10.1207/s15327744joce1403_01.

[10] S. Gupta *et al.*, ‘Evaluating Waterfall vs. Agile Models in Software Development for Efficiency and Adaptability’, in *Advances in Logistics, Operations, and Management Science*, IGI Global, 2024, pp. 142–148. doi: 10.4018/979-8-3693-3318-1.ch008.

[11] A. Mishra and Y. I. Alzoubi, ‘Structured software development versus agile software development: a comparative analysis’, *Int J Syst Assur Eng Manag*, vol. 14, no. 4, pp. 1504–1522, Aug. 2023, doi: 10.1007/s13198-023-01958-5.

[12] A. Moran, *Agile Risk Management*. in SpringerBriefs in Computer Science. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-05008-9.

[13] J. Nyfjord and M. Kajko-Mattsson, ‘Commonalities in Risk Management and Agile Process Models’, in *International Conference on Software Engineering Advances (ICSEA 2007)*, Cap Esterel, France: IEEE, Aug. 2007, pp. 18–18. doi: 10.1109/ICSEA.2007.22.

[14] M. Hammad, I. Inayat, and M. Zahid, ‘Risk Management in Agile Software Development: A Survey’, in *2019 International Conference on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan: IEEE, Dec. 2019, pp. 162–1624. doi: 10.1109/FIT47737.2019.00039.

[15] A. Agrawal, Mohd. A. Atiq, and L. S. Maurya, ‘A Current Study on the Limitations of Agile Methods in Industry Using Secure Google Forms’, *Procedia Computer Science*, vol. 78, pp. 291–297, 2016, doi: 10.1016/j.procs.2016.02.056.

[16] B. Boehm, ‘Balancing Agility and Discipline: A Guide for the Perplexed’, in *Software Engineering Research and Applications*, vol. 3026, C. V. Ramamoorthy, R. Lee, and K. W. Lee, Eds., in Lecture Notes in Computer Science, vol. 3026. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–1. doi: 10.1007/978-3-540-24675-6_1.
[17] M. H. Zahedi, A. Rabiei Kashanaki, and E. Farahani, ‘Risk management framework in Agile software development methodology’, *IJECE*, vol. 13, no. 4, p. 4379, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4379-4387.
[18] O. M. Thom-Manuel, ‘Explicit Risk Management in Agile Software Projects: Its Relevance and Benefits’, *AJRCoS*, pp. 12–24, Aug. 2022, doi: 10.9734/ajrcos/2022/v14i330340.
[19] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA: Pearson, 2016.
[20] G. Low and V. Leenanuraksa, ‘Software quality and CASE tools’, in *STEP ’99. Proceedings Ninth International Workshop Software Technology and Engineering Practice*, Pittsburgh, PA, USA: IEEE Comput. Soc, 1999, pp. 142–150. doi: 10.1109/STEP.1999.798787.
[21] G. Pascarella *et al.*, ‘Risk Analysis in Healthcare Organizations: Methodological Framework and Critical Variables’, *RMHP*, vol. Volume 14, pp. 2897–2911, Jul. 2021, doi: 10.2147/RMHP.S309098.
[22] A. Vaezi, S. Jones, and A. Asgary, ‘Integrating Resilience into Risk Matrices: A Practical Approach to Risk Assessment with Empirical Analysis’, *JRACR*, vol. 13, no. 4, Jan. 2024, doi: 10.54560/jracr.v13i4.411.
[23] R. Khan, ‘The Role of Risk Management in Project Success: A Comprehensive Study of Project Management Practices’, *TIJMG*, vol. 11, no. 01, Feb. 2025, doi: 10.21522/TIJMG.2015.11.01.Art015.
[24] S. Moran, ‘How to Make Sure Your Design Is Reasonably Safe and Sustainable’, in *An Applied Guide to Process and Plant Design*, Elsevier, 2015, pp. 217–245. doi: 10.1016/B978-0-12-800242-1.00015-3.
[25] N. C. Bradley, T. Fritz, and R. Holmes, ‘Sources of software development task friction’, *Empir Software Eng*, vol. 27, no. 7, p. 175, Dec. 2022, doi: 10.1007/s10664-022-10187-6.
[26] H. Saeeda, M. O. Ahmad, and T. Gustavsson, ‘Challenges in Large-Scale Agile Software Development Projects’, in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, Tallinn Estonia: ACM, Mar. 2023, pp. 1030–1037. doi: 10.1145/3555776.3577662.
[27] A. Mihalache, ‘Project Management Tools for Agile Teams’, *IE*, vol. 21, no. 4/2017, pp. 85–93, Dec. 2017, doi: 10.12948/issn14531305/21.4.2017.07.
[28] S. Chaouch, A. Mejri, and S. A. Ghannouchi, ‘A framework for risk management in Scrum development process’, *Procedia Computer Science*, vol. 164, pp. 187–192, 2019, doi: 10.1016/j.procs.2019.12.171.
[29] B. W. Boehm, ‘Software risk management: principles and practices’, *IEEE Softw.*, vol. 8, no. 1, pp. 32–41, Jan. 1991, doi: 10.1109/52.62930.
[30] R. P. Higuera and Y. Y. Haimes, ‘Software Risk Management’, p. 1996.
[31] J. Kontio, *Risk Management in Software Development: A Technology Overview and the Riskit Method*. 2001. [Online]. Available: https://insights.sei.cmu.edu/library/capability-maturity-model-integration-cmmi-version-12-overview/
[32] B. G. Tavares, M. Keil, C. E. Sanches Da Silva, and A. D. De Souza, ‘A Risk Management Tool for Agile Software Development’, *Journal of Computer Information Systems*, vol. 61, no. 6, pp. 561–570, Nov. 2021, doi: 10.1080/08874417.2020.1839813.
[33] V. Anes, A. Abreu, and R. Santos, ‘A New Risk Assessment Approach for Agile Projects’, in *2020 International Young Engineers Forum (YEF-ECE)*, Costa da Caparica, Portugal: IEEE, Jul. 2020, pp. 67–72. doi: 10.1109/YEF-ECE49388.2020.9171808.
[34] T. Natarajan and S. Pichai, ‘Behaviour-driven development and metrics framework for enhanced agile practices in scrum teams’, *Information and Software Technology*, vol. 170, p. 107435, Jun. 2024, doi: 10.1016/j.infsof.2024.107435.

[35] L. Barros, C. Tam, and J. Varajão, 'Agile software development projects–Unveiling the human-related critical success factors', *Information and Software Technology*, vol. 170, p. 107432, Jun. 2024, doi: 10.1016/j.infsof.2024.107432.
[36] M. Shafir, P. P. Saha, A. T. Araf, J. F. Nishi, M. Hasan, and F. Sadia, 'The Success Factors of Agile Methodologies in Software Development based on Developing Countries' Software Firms', *Procedia Computer Science*, vol. 256, pp. 1954–1961, 2025, doi: 10.1016/j.procs.2025.02.338.
[37] E. Karimian Sichani, A. Smith, K. El Emam, and L. Mosquera, 'Creating High-Quality Synthetic Health Data: Framework for Model Development and Validation', *JMIR Form Res*, vol. 8, p. e53241, Apr. 2024, doi: 10.2196/53241.
[38] E. Barbierato, M. L. D. Vedova, D. Tessera, D. Toti, and N. Vanoli, 'A Methodology for Controlling Bias and Fairness in Synthetic Data Generation', *Applied Sciences*, vol. 12, no. 9, p. 4619, May 2022, doi: 10.3390/app12094619.
[39] PMI, 'A guide to the project management body of knowledge (PMBOK® guide) (7th ed.)'. Project Management Institute, 2021.
[40] K. C. Obondia, 'The utilization of project risk monitoring and control practices and their relationship with pro-ject success in construction projects', *Journal of Project Management,* vol. 7, no. 1, pp. 35–25, 2022.
[41] S. Kainulainen, T. Tuunanen, and T. Vartiainen, 'Requirements Risk Management for Continuous Development: Organisational Needs', *AJIS*, vol. 28, Sep. 2024, doi: 10.3127/ajis.v28.4441.
[42] ISACA, 'Agile Methodologies Require Adjustments to Risk Management'. ISACA, 2023. [Online]. Available: https://www.isaca.org/resources/news-and-trends/newsletters/atisaca/2023/volume-7/agile-methodologies-require-adjustments-to-risk-management