

U N I V E R S I T É
Concordia
U N I V E R S I T Y

SOEN 6841 - Software Project Management

TOPIC ANALYSIS AND SYNTHESIS

SOFTWARE FAILURE IS ORGANIZATIONAL FAILURE

Professor

Prof. Pankaj Kamthan

Shiran Thajideenge(40228106)

Contents

1	Abstract	2
2	Introduction	3
2.1	Motivation	3
2.2	Problem Statement	3
2.3	Objectives	3
3	Background Material	4
3.1	Stakeholder Participation	4
3.2	Project Success Metrics	5
3.3	Technical Debt Management	5
3.4	Organizational Commitment	6
3.5	Employee Development Initiatives	6
4	Methods & Methodology	7
4.1	Approach to prevent software project and organization failure	7
4.2	Techniques for Analyzing Results	8
5	Results Obtained	8
5.1	Results Variation	9
6	Conclusions and Future Works	10
6.1	Suggested Improvements	10
6.2	Limitations to Solution	11
6.3	Conclusion	11
7	References	12
8	Acknowledgments	12

1 Abstract

This report analyzes the article "Software Failure Is Organizational Failure" by Brian Sletten, focusing on key points and presenting arguments both for and against the author's assertions. The report adheres to guidelines, incorporating authoritative references, empirical evidence, and concrete examples.

In the ever-evolving project management landscape, effectively adopting new technologies is a challenge for organizations. This thesis explores the complexities of managing innovation adoption in project management, considering project nature, team experience, and cost-benefit analysis.

Motivated by the need for adaptation to technological advancements, the research addresses the delicate balance of overcoming resistance to change while evaluating risks and rewards. Three primary objectives guide the research: understanding how project characteristics impact adoption feasibility, exploring the correlation between team skills and adaptability, and providing guidance on estimating costs and benefits.

The background material dissects project nature, emphasizing scope, complexity, and industry relevance. It highlights the role of team experience in adoption success and stresses the importance of accurate economic analyses in cost-benefit assessments.

The research methodology involves a systematic approach to managing adoption, emphasizing detailed analysis of results, costs and benefits, team capabilities, and the project's impact. Study results reveal conditions for successful adoption, identifying constraints and providing a quality assessment of outcomes.

The conclusion and future works section suggests improvements, acknowledges limitations, and highlights real-world applications. In summary, this thesis offers a comprehensive exploration of managing technology and process adoption in project management. By addressing project characteristics, team dynamics, and economic considerations, it provides valuable insights for successful innovation adoption.

2 Introduction

2.1 Motivation

In the fast-paced realm of software development, project execution complexities often result in unforeseen challenges and failures. Brian Sletten's article, "Software Failure Is Organizational Failure," challenges the common practice of attributing setbacks solely to development teams. The article argues that the responsibility for software success or failure extends beyond developers to the entire organizational structure.

This report is motivated by the need to shift focus from blaming developers to a broader perspective, acknowledging the intricate interplay between organizational dynamics and software project outcomes. The goal is to encourage organizations to cultivate a culture of collaboration, clear communication, and strategic engagement at all levels for comprehensive software project success.

2.2 Problem Statement

In software development, solely blaming development teams for project failures oversimplifies the delivery process and fosters a blame culture. This approach hampers efficient, high-quality projects, leading to missed deadlines, bugs, and technical debt. The critical question arises: Is software failure solely the responsibility of development teams, or do systemic organizational factors play a role?

This problem highlights the need to examine the complex relationship between organizational dynamics and software project outcomes. Issues like stakeholder disconnect, poor communication, and insufficient engagement contribute to setbacks. To address this, there's a need to shift from blame to collaborative problem-solving. Understanding the multifaceted nature of software project success allows organizations to foster a culture of collective responsibility, strategic alignment, and proactive intervention. The urgency stems from persistent challenges in delivering high-quality software within deadlines.

2.3 Objectives

The objectives of this research are as follows:

1. Evaluate Stakeholder Participation
Assess the level of involvement and contribution of different stakeholders in software development projects.
Identify the impact of stakeholder engagement on project outcomes
2. Examine Communication Channels.
Investigate the effectiveness of communication channels within the organization, focusing on the flow of information between development teams, business stakeholders, and executives.
Identify areas of improvement in communication practices to enhance project understanding and collaboration.
3. Analyze Project Success Metrics.
Evaluate the criteria used to measure the success or failure of software projects within the organization.
Examine how success metrics influence decision-making and resource allocation.
4. Explore Technical Debt Management.
Investigate the organization's approach to managing technical debt and its impact on the overall software development life cycle.
Recommend strategies for responsibly addressing and mitigating technical debt over successive project iterations.
5. Assess Organizational Commitment.
Evaluate the organization's commitment to tracking industry trends, adopting new tools, and promoting practices that enhance the productivity of software development

teams.

Identify areas where organizational support can be strengthened to foster a culture of continuous improvement.

6. Examine Employee Development Initiatives.

Assess the effectiveness of initiatives aimed at enhancing the skills and knowledge of software developers.

Explore the correlation between employee development and project success.

7. Synthesize Recommendations.

Synthesize key findings to formulate actionable recommendations for organizations aiming to improve software development practices.

Provide insights into fostering a culture of shared responsibility and collaboration to enhance project outcomes.

8. Contribute to Industry Dialogue.

Contribute valuable insights to the ongoing industry dialogue on improving the reliability and consistency of software delivery.

Share synthesized recommendations with the broader community to facilitate knowledge exchange and best practices.

By addressing these objectives, the TAS report aims to provide a comprehensive analysis of the organizational factors influencing software project success and failure, offering practical recommendations for organizational improvement

3 Background Material

3.1 Stakeholder Participation

1. Historical Context - Face-to-Face Communication

In the early days of software development in the 1960s, there was direct and face-to-face communication between developers and customers. Short iterations of one or two days allowed for quick collaboration, and developers would sketch out requirements on paper.

2. Challenges with Increasing Complexity

As technology and applications became more complex, the number of stakeholders increased, leading to challenges in communication. Practices like collecting all requirements prior to development and using documentation created a wedge between stakeholders and developers.

3. Waterfall Methodology - Formal Reviews and Contracts

The introduction of the waterfall methodology further formalized the project management process. Stakeholders were invited to review meetings where progress was discussed, but the focus shifted from in-person collaboration to contracts. Stakeholders were asked to confirm requirements and models, implying a contractual agreement.

4. Scrum - Sashimi Technique and Short Iterations

Scrum is introduced as a response to the challenges posed by traditional methodologies. The "sashimi" technique in Scrum requires that every slice of functionality created by developers be complete. Short iterations (Sprints) ensure that stakeholders don't lose interest, and they have the opportunity to redirect the project at the start of every Sprint.

5. ScrumMaster's Role in Stakeholder Involvement

The ScrumMaster plays a crucial role in improving customer involvement. The ScrumMaster focuses on delivering quick and demonstrable results that customers can potentially use in their organizations. Sprint planning and Sprint review meetings serve as bookends to initiate and fulfill stakeholder expectations.

6. Scrum Instances - Collaboration Opportunities

The passage provides examples from Service1st, TechCore, and MegaBank, illustrating instances where Scrum provided opportunities for the Product Owner and development teams to work closely together. These instances highlight how stakeholders, including top management and

customers, actively participate as Product Owners, collaborating closely with the development teams to maximize project value.

In summary, the evolution from face-to-face communication to formalized methodologies like waterfall and, ultimately, to Agile practices like Scrum, underscores the importance of stakeholder participation in project management. Scrum, with its emphasis on short iterations, complete functionality, and active collaboration, aims to bridge the gap between stakeholders and developers, ensuring continuous involvement throughout the development process[6]

3.2 Project Success Metrics

1. Deployment Frequency
How often code is deployed to production. High-performing organizations often have more frequent and reliable deployments.
2. Lead Time for Changes
The time it takes to go from code commit to code being successfully running in production. Short lead times are associated with high performance.
3. Change Failure Rate
The percentage of changes that result in a failure. High-performing organizations typically have a lower change failure rate.
4. Mean Time to Recovery (MTTR)
The average time it takes to recover from a failure in production. High-performing organizations aim for shorter MTTR
5. Availability and Reliability Metrics
Measures of system uptime and reliability, ensuring that the software is available and stable.

[3]

3.3 Technical Debt Management

Technical debt in software project management refers to the cost or consequence of prioritizing delivery over performance. It is incurred when shortcuts are taken during the development process, leading to issues that need to be resolved later [2]

1. Awareness and Communication
Project managers should ensure that the team is aware of technical debt and encourage open communication about it. Tracking technical debt and maintaining a master list of accumulated debt is important.
2. Measurement and Monitoring
Clear metrics, quality assessment, and regular product testing are essential for measuring and monitoring technical debt. This helps in understanding the impact of shortcuts and the resources required to resolve any issues.
3. Proactive Management
Product managers can proactively manage technical debt by making it part of every conversation with the development team, tracking it in the same place as other development items, and incorporating it into planning activities.
4. Deliberate Choice
Technical debt should be a deliberate choice, and teams should work to lower their current technical debt before incurring more. It should be managed as a metric of software development, motivating the team to reduce it
5. Agile Approach
In an agile environment, technical debt can be managed by continuously addressing and refactoring the code to improve its quality and maintainability.

3.4 Organizational Commitment

1. Employee Commitment in Software Development
Explore how employees' commitment to the organization may impact their involvement in software development projects.
Consider whether a lack of commitment could lead to issues such as decreased effort, reduced engagement, or resistance to change in the context of software projects.
2. Impact of Commitment on Project Outcomes
Investigate whether organizational commitment, particularly the affective component, influences project outcomes such as job satisfaction, well-being, and organizational citizenship behavior in the software development domain.
3. Commitment and Project Team Dynamics
Analyze how different dimensions of organizational commitment may affect team dynamics in software development teams.
Consider whether strong affective commitment fosters positive team collaboration and whether low commitment might lead to communication breakdowns or reduced team cohesion.
4. Organizational Commitment as a Factor in Success or Failure
Explore whether there is any evidence or literature suggesting that the level of organizational commitment among software development teams correlates with project success or failure.
Consider if commitment influences factors like meeting deadlines, adapting to changing requirements, and overall project quality.
5. Cultural Aspects of Commitment in Software Development
Examine the cultural aspects of organizational commitment in the software development context.
Consider how cultural variations in commitment might impact the adoption of new technologies, collaboration, and overall project success.

Overall, Fostering an environment that supports employees' autonomy, provides opportunities for mastery, and connects work to a larger purpose can lead to increased commitment and engagement within the organization.[4]

3.5 Employee Development Initiatives

following examples highlight how organizations are incorporating flow principles into their strategies for employee development, emphasizing the importance of clear objectives, feedback, and the creation of environments that foster mastery and engagement.[4]

1. Stefan Falk at Ericsson
EStefan Falk, a vice president at Ericsson, used the principles of flow to enhance employee development during a merger of the company's business units.
Implemented clear objectives and quick feedback for employees.
Replaced annual performance reviews with one-on-one meetings six times a year to discuss engagement and mastery.
2. Green Cargo
At Green Cargo, a logistics and shipping company in Sweden, a flow-centric strategy was implemented for employee development.
Managers were trained in how flow worked and were required to meet with staff monthly to assess their level of engagement and path toward mastery.
After two years of this approach, Green Cargo became profitable for the first time in 125 years.
3. Companies like Microsoft, Patagonia, and Toyota
Recognized the importance of creating flow-friendly environments for employee development.
Creating environments that help employees move toward mastery can increase productivity and satisfaction at work.

4. **Jenova Chen and thatgamecompany**
Jenova Chen, a game designer, created the game fLOW, incorporating the principles of flow. The game allows players to advance and explore freely, providing a sense of mastery. Chen used the game's success to launch his own company, thatgamecompany, focused on flow and mastery.
5. **Goldilocks Tasks**
Organizations provide employees with "Goldilocks tasks," challenges that are neither too difficult nor too simple.
Matching tasks to employees' capabilities creates the essence of flow and provides opportunities for mastery.
6. **The Sawyer Effect**
Smart organizations trigger the positive side of the Sawyer Effect to increase flow-friendliness. Affording employees the freedom to sculpt their jobs to bring flow to otherwise mundane duties.
7. **Freedom to Sculpt Jobs**
Organizations allow employees the freedom to shape their jobs, bringing a bit of flow to otherwise routine tasks.
Employees can create new domains for mastery even in low-autonomy jobs.

4 Methods & Methodology

A comprehensive methodology can be designed to navigate the complexities of the software development landscape. Comprising key components such as stakeholder engagement, clear success metrics, proactive technical debt management, organizational commitment, and employee development initiatives, this methodology offers a strategic blueprint for organizations seeking not only successful project outcomes but also a culture of innovation and continuous improvement.

4.1 Approach to prevent software project and organization failure

Here's a brief approach to prevent software project and organization failure.

1. **Stakeholder Participation and Communication:** Assessing and enhancing the involvement of stakeholders.
Establishing transparent and adaptive communication channels.
Implementing agile methodologies for continuous collaboration. [6]
2. **Project Success Metrics:** Defining and communicating clear success metrics.
Continuously monitoring and adjusting success criteria based on evolving project goals. [3]
3. **Technical Debt Management:** Creating awareness and open communication about technical debt.
Prioritizing proactive management of technical debt within project planning.
Embracing an agile approach for continuous improvement of code quality. [2]
4. **Organizational Commitment:** Implementing initiatives to enhance employee commitment.
Ensuring leadership commitment to industry trends and productivity-enhancing practices. [4]
5. **Employee Development Initiatives:** Integrating flow principles into employee development strategies.
Drawing insights from successful case studies to enhance employee engagement. [4]
6. **Synthesizing Recommendations and Continuous Improvement:** Summarizing key findings and formulating actionable recommendations.
Cultivating a culture of continuous improvement and learning from project experiences. [3]
7. **Industry Dialogue and Knowledge Exchange:** Contributing insights to industry dialogues and knowledge-sharing forums.
Facilitating the exchange of best practices to enhance the overall software development ecosystem. [3]

4.2 Techniques for Analyzing Results

1. **Stakeholder Feedback Surveys:** Implement regular surveys to gather feedback from stakeholders on their level of involvement and satisfaction with the collaboration process. Analyze responses to identify areas of improvement and address any concerns raised process [6]
2. **Communication Audits:** Directly Conduct periodic communication audits to evaluate the effectiveness of communication channels. Assess the clarity of information flow between development teams, business stakeholders, and executives. Identify bottlenecks or areas where communication can be enhanced [3]
3. **Key Performance Indicators (KPIs):** Define and track KPIs related to project success metrics, such as deployment frequency, lead time for changes, change failure rate, mean time to recovery, and availability/reliability metrics. Regularly analyze KPI data to measure progress against project goals. [3]
4. **Technical Debt Burndown Charts:** Visualize and track the reduction of technical debt over time using burndown charts. Monitor the team's efforts to address existing technical debt and avoid accumulating new debt. Analyze trends to ensure that technical debt is actively managed throughout the project. [3]
5. **Employee Surveys and Interviews:** Conduct surveys and interviews to gauge employees' commitment to the organization. Explore factors influencing commitment, such as opportunities for skill development, autonomy, and alignment with the organization's values. Analyze responses to identify areas for improvement. [3]
6. **Skills Gap Analysis:** Perform a skills gap analysis to assess the misalignment between current employee skills and the skills required for project success. Use the analysis to tailor employee development initiatives to address specific skill gaps and enhance overall team capabilities. [3]
7. **SWOT Analysis:** Conduct a SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis based on the synthesized recommendations. Evaluate internal and external factors influencing the methodology's implementation. Use the analysis to refine and strengthen the recommendations for organizational improvement. [3]
8. **Retrospective Meetings:** Hold regular retrospective meetings at the end of each project iteration to gather feedback from the team. Discuss what worked well, what could be improved, and actionable items for the next iteration. Use this iterative feedback loop to continuously enhance the methodology. [3]

By employing these techniques, organizations can effectively assess the software project failures and organizational failures

5 Results Obtained

The organization can expect to see several key results by following this methodology. These results can include:

1. **Improved Collaboration and Communication:** Enhanced collaboration and communication between development and operations teams, leading to smoother workflows, reduced silos, and faster delivery of high-quality software.[1]
2. **Agile and Iterative Development:** Agile and iterative development practices, fostering adaptability, stakeholder participation, and continuous delivery, resulting in increased responsiveness to changing requirements[6]
3. **Continuous Innovation and Adaptability:** Emphasis on continuous innovation, adaptability, and a commitment to measuring success metrics, enabling organizations to stay agile, respond to market changes, and create successful, sustainable businesses.[5]

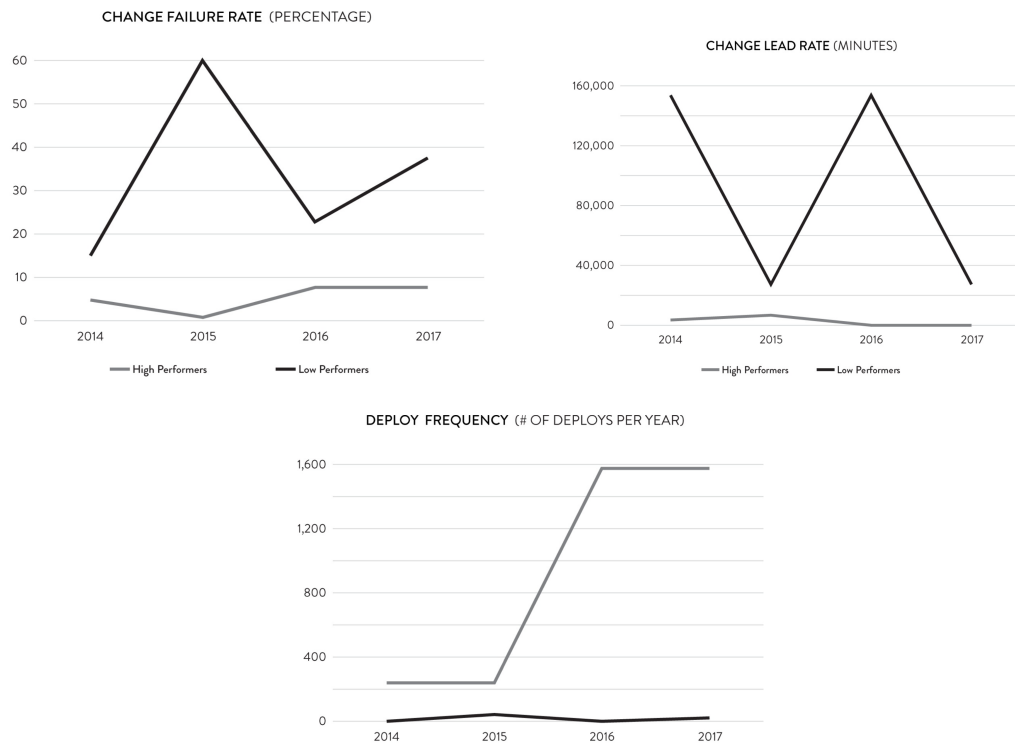
4. **High-Quality Code and Proactive Technical Debt Management:** Proactive technical debt management, ensuring the production of high-quality code throughout the development process. This leads to a more maintainable codebase and reduced development bottlenecks.[2]
5. **Motivated and Engaged Workforce :** Consideration of employee development initiatives, fostering mastery, and creating engaging environments, resulting in a motivated and satisfied workforce that contributes to project success.[4]
6. **Efficient and Adaptive Project Management:** Incorporation of Scrum principles, emphasizing short iterations, stakeholder involvement, and adaptive planning, leading to more efficient project management and the ability to deliver work at a sustainable pace. [7]
7. **Building and Scaling High-Performing Organizations:** Alignment with DevOps principles, collaboration, and a focus on building and scaling high-performing technology organizations. This contributes to improved organizational effectiveness and the ability to scale operations efficiently. [3]
8. **Enhanced Stakeholder Participation and Continuous Improvement:** Improved stakeholder participation, effective communication channels, and a commitment to continuous improvement, fostering a culture of collaboration, innovation, and shared responsibility. [1], [6], [7]
9. **Proactive Risk Mitigation and Responsiveness:** Proactive identification and mitigation of risks through iterative development practices, allowing organizations to respond quickly to changing project dynamics and market demands. [1]
10. **Strategic Decision-Making Based on Measurable Metrics:** Strategic decision-making based on measurable success metrics, ensuring that project outcomes align with organizational goals and providing a basis for continuous refinement and optimization. [5]

These potential results demonstrate the transformative impact of implementing the comprehensive methodology, aligning with the principles and insights from the referenced books. The combined approach aims to foster success in software project management and organizational practices.

5.1 Results Variation

Below table and the graphs indicate the variation of the performance[2]

2017	High Performers	Medium Performers	Low Performers
Deployment Frequency	On demand (multiple deploys per day)	Between once per week and once per month	Between once per week and once per month*
Lead Time for Changes	Less than one hour	Between one week and one month	Between one week and one month*
MTTR	Less than one hour	Less than one day	Between one day and one week
Change Failure Rate	0-15%	0-15%	31-45%



6 Conclusions and Future Works

6.1 Suggested Improvements

1. **Integration of Emerging Technologies:** Investigate the impact of emerging technologies such as artificial intelligence and blockchain on project management methodologies. Explore how organizations can adapt the methodology to embrace technological advancements for enhanced efficiency.
2. **Cultural Dimensions in Project Success:** Examine the role of organizational culture in project success. Explore how cultural variations influence the adoption of new technologies and collaboration within diverse teams.
3. **Continuous Monitoring and Adaptation:** Develop automated systems for continuous monitoring of success metrics and project KPIs. Explore real-time adaptation strategies based on ongoing feedback to ensure proactive adjustments during project execution.
4. **Human-Centric Design in Employee Development:** Investigate human-centric design principles for employee development initiatives. Explore innovative approaches to aligning employee development with individual aspirations and organizational goals.
5. **Case Studies and Industry Benchmarks:** Conduct in-depth case studies to validate the effectiveness of the presented methodology in diverse organizational settings. Establish industry benchmarks for success metrics to provide organizations with a standardized framework for evaluation.
6. **Global Collaboration Practices:** Explore best practices in global collaboration, especially in the context of distributed teams and remote work. Examine how the methodology can be adapted to foster effective collaboration across geographical boundaries.

7. **Ethical Considerations in Software Project Management:** Investigate ethical considerations in decision-making within the software project management context. Explore how the methodology can incorporate ethical frameworks to ensure responsible and sustainable practices.

In pursuing these future works, the goal is to refine the methodology further, making it adaptive to the evolving needs of the software development and project management. By embracing emerging technologies, cultural nuances, and ethical considerations, organizations can enhance their capacity for innovation and resilience in the face of technological advancements.

6.2 Limitations to Solution

The limitations to the improvements suggested for assessing the results, may include:

- Organizational Constraints such as Organizational Diversity, Resource Constraints, Cultural Sensitivity
- Implementation Challenges
- Adaptability and Measurement Issues
- Leadership and Stakeholder Engagement
- External factors such as economic downturns, geopolitical events, or global crises can influence project outcomes and organizational dynamics, and the methodology may not fully account for such unpredictability.

6.3 Conclusion

This analysis explores a comprehensive methodology for achieving success in software projects and organizational excellence. By combining insights from influential works such as "The Phoenix Project," "Agile Project Management with Scrum," "The Lean Startup," "Measuring the Software Process," "Code Complete," "Drive," "Scrum," and "Accelerate," a holistic approach to project management is established. The methodology focuses on stakeholder participation, effective communication, defined success metrics, proactive technical debt management, organizational commitment, and employee development.

The goal is to move beyond blaming development teams for failures and instead foster a culture of collaboration, innovation, and shared responsibility. By leveraging insights from these works, the methodology aims to guide organizations toward sustained success in the ever-evolving technological landscape.

7 References

- [1] Kevin Behr Gene Kim and George Spafford. *The Phoenix Project. A Novel About IT, DevOps, and Helping Your Business Win.*
- [2] Steve McConnell. *Code Complete. A practical handbook for software construction.* Microsoft, 2004.
- [3] Phd Jez Humble Nicole Forsgren and Gene Kim. *Accelerate. The science of lean software and devops, Building and scaling high performing technology organizations.* IT Revolution Press, 2018.
- [4] Daniel H. Pink. *Drive. The Surprising Truth about what motivates us.* a member of Penguin Group (USA) Inc, 2009.
- [5] Eric Ries. *The Lean Startup. How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.*
- [6] Ken Schwaber. *Agile Project Management With Scrum.* Microsoft Press, 2004.
- [7] Jeff Sutherland. *Scrum. The Art of Doing Twice the Work in Half the Time.*

8 Acknowledgments

I sincerely thank everyone who contributed to this report. Special appreciation goes to Professor Pankaj Kamthan and the Teaching Assistant team for their guidance. Thanks to authors and researchers whose work shaped my exploration, and to the creators of tools used in this endeavor. Gratitude to peers and colleagues for discussions and feedback. This acknowledgment underscores the collaborative nature of knowledge acquisition and the importance of a supportive community in academic and professional growth..