Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 3, Issue 9, 2025

AUTOMATING AGILE DECISIONS: A COMPARATIVE ANALYSIS OF AI-POWERED REQUIREMENT PRIORITIZATION TOOLS

Shawaiz Arif[*1], Neha Ijaz[2], Muhammad Faisal[3], Muhammad Ahmed[4], Amna Khan[5]

[*1]*Faculty of Computer Science & Information Technology, Superior University, Pakistan*
[2]*Computing International Qualifications (IQ), University of Lahore (UOL), Pakistan*
[3]*Department of Computer Science and Engineering, University of Hafr AlBatin, Saudi Arabia*

[*1]shawaizarif1@gmail.com, [2]nehaijaz@gmail.com, [3]faisalshafiq02@outlook.com,
[4]ahmadkahloon@superior.edu.pk, [5]amna.khan@uhb.edu.sa

DOI: https://doi.org/10.5281/zenodo.17097664

Keywords

Article History
Received: 19 June 2025
Accepted: 29 August 2025
Published: 11 September 2025

Corresponding Author: *
Shawaiz Arif

Abstract

Requirement prioritization is a cornerstone of Agile software engineering, yet traditional methods such as MoSCoW and the Analytical Hierarchy Process (AHP) often struggle with scalability, subjectivity, and stakeholder bias. Recent advances in Artificial Intelligence (AI), including natural language processing (NLP), machine learning (ML), and sentiment analysis, offer opportunities to automate and enhance this process.

This study presents a comparative analysis of prominent AI-powered requirement prioritization tools, evaluating them across four dimensions: AI capabilities, cost structures, integration with Agile ecosystems, and usability. Data was drawn from product documentation, user reviews, academic literature, and expert insights. Tools such as Airfocus, Craft.io, ClickUp AI, ReqSuite RM, Kanoah Tests, Zepel, and Aha! Roadmaps were systematically assessed.

Findings reveal that no single tool is universally optimal; instead, suitability depends on organizational scale, resource availability, and integration needs. While some platforms excel in predictive analytics and compliance, others emphasize affordability or ease of onboarding. However, gaps remain in transparency and explainability, limiting stakeholder trust in AI-driven outputs. This paper contributes by offering both a practical guide for Agile teams and a research-oriented discussion of adoption barriers, ethical considerations, and future directions. As AI-driven prioritization matures, the integration of explainable AI (XAI), hybrid human-AI decision-making, and emerging techniques such as large language models (LLMs) will be critical for advancing requirement engineering practices.

Index Terms– Requirement prioritization, Artificial intelligence (AI), Agile software engineering, Requirements engineering (RE), Natural language processing (NLP), Explainable AI (XAI), Tool comparison, Machine learning (ML), Human-AI collaboration, Cost-benefit analysis.

INTRODUCTION

REQUIREMENT prioritization is a critical activity in software engineering and has been identified as a determinant of project success, particularly in Agile environments where requirements evolve rapidly and development cycles are time-boxed [1], [2]. Agile methodologies emphasize continuous delivery and stakeholder collaboration, which necessitate systematic approaches to selecting and implementing the most valuable requirements within each iteration.

Traditional prioritization approaches, such as MoSCoW, the Analytical Hierarchy Process (AHP), the $100 method, and the Kano model, have been widely applied to support decision-making. While these methods provide structured mechanisms for ranking requirements, they are often criticized for their limited scalability, high subjectivity, and susceptibility to stakeholder bias in large and complex projects [2], [18], [19]. Furthermore, empirical investigations have highlighted barriers to the adoption of these techniques, including lack of traceability, cognitive overload, and inconsistent use across organizations [10].

Recent advances in Artificial Intelligence (AI) provide opportunities to overcome these limitations. Techniques such as natural language processing (NLP), sentiment analysis, and supervised machine learning enable the automated analysis of requirements, user stories, and stakeholder feedback. Such approaches facilitate the generation of prioritized backlogs with reduced manual effort and improved objectivity [3], [4], [11], [12], [14], [15], [17]. AI-driven prioritization not only improves efficiency and scalability but also supports data-driven decision-making, which is increasingly central to digital transformation initiatives [14].

Despite these advancements, challenges remain in ensuring the interpretability and trustworthiness of AI-generated results. The opacity of complex AI models raises concerns regarding stakeholder confidence and acceptance, particularly in regulated domains where accountability is critical [13], [20], [21]. These concerns highlight the need for explainable AI (XAI) and hybrid human–AI collaboration to ensure effective adoption in practice.

This study addresses the identified gap by presenting a comparative analysis of AI-powered requirement prioritization tools. The evaluation considers four dimensions: AI capabilities, cost and licensing, integration within Agile ecosystems, and usability. Data were collected from product documentation, empirical user reviews, academic literature, and practitioner insights to ensure a balanced and systematic comparison. The objective is to provide both practitioners and researchers with evidence-based guidance for selecting tools that align with organizational needs, resource constraints, and maturity levels.

Background and Related Work

Requirements engineering (RE) encompasses the processes of eliciting, documenting, validating, and prioritizing software requirements, serving as a foundational activity in the software development lifecycle [1]. Within Agile methodologies, requirement prioritization is particularly significant, as iterative and time-boxed delivery models necessitate the continual selection of high-value requirements that align with stakeholder needs and project objectives [2].

Traditional prioritization techniques, including the MoSCoW method, the Analytical Hierarchy Process (AHP), the $100 method, and the Kano model, have been widely employed in practice. These techniques provide structured approaches to decision-making but suffer from several limitations. They are inherently labor-intensive, prone to stakeholder subjectivity, and lack scalability in large or distributed projects [2], [18], [19]. Moreover, empirical investigations indicate that many organizations struggle with their adoption due to barriers such as stakeholder conflicts, absence of traceability, and cognitive overhead [10].

In response to these challenges, Artificial Intelligence (AI) has been increasingly explored as a means of enhancing requirement prioritization. Techniques such as natural language processing (NLP), supervised machine learning, and sentiment analysis allow for the automated processing of requirement specifications, user stories, and stakeholder feedback. These approaches enable the generation of prioritization outcomes that are more consistent, scalable, and data-driven than manual methods [11], [12], [14], [15], [17]. For example, sentiment-aware models have been shown to infer requirement importance from stakeholder commentary [16], while hybrid methods combine rule-based reasoning with predictive analytics to balance accuracy with interpretability [14], [17].

Despite these advances, significant research gaps remain. Commercial platforms increasingly advertise AI-enhanced prioritization features, yet there is a scarcity of systematic, tool-based comparative analyses in the literature [11]. Furthermore, concerns regarding the transparency and explainability of AI-driven results limit their acceptance in professional practice, especially in regulated or safety-critical domains [13], [20], [21]. These limitations underscore the need for empirical studies that bridge academic approaches and industrial implementations, while also addressing ethical and organizational considerations surrounding the deployment of AI in requirements engineering.

Literature Review

The evolution of requirement prioritization has been shaped by both academic inquiry and industrial practice. Early research in requirements engineering emphasized structured prioritization frameworks such as MoSCoW, the Analytical Hierarchy Process (AHP), and pairwise comparisons to support decision-making [1], [2], [18], [19]. These methods provided valuable mechanisms for achieving consensus among stakeholders; however, they were found to lack scalability, suffer from subjectivity, and impose significant cognitive overhead in dynamic Agile environments [2], [10].

In response to these limitations, researchers have investigated the application of Artificial Intelligence (AI) to requirements engineering. Machine learning (ML), natural language processing (NLP), and sentiment analysis have been employed to automate prioritization and reduce reliance on manual judgment [11], [12], [14], [15], [17]. For instance, sentiment-aware models have been developed to assign priorities based on the emotional intensity and contextual relevance of stakeholder feedback [16], while hybrid approaches have combined rule-based reasoning with predictive analytics to balance interpretability with performance [14]. These AI-enabled methods enhance scalability, reduce bias, and support data-driven decision-making in Agile software development contexts.

Another important development in the literature is the emphasis on explainable AI (XAI). While ML models are capable of producing optimized prioritization outcomes, their opacity raises concerns regarding stakeholder trust and organizational adoption [13], [20]. Studies have shown that without adequate transparency, stakeholders may resist AI-driven recommendations, especially in regulated domains where accountability is paramount [21]. This has led to calls for hybrid human–AI approaches, where algorithmic recommendations are complemented by stakeholder validation [14], [20].

Despite these advances, there remains a gap between academic research and practical tool implementation. While individual studies propose AI-based prioritization models, few have systematically compared commercially available tools or evaluated their adoption in real-world Agile projects [11]. Furthermore, empirical benchmarks for accuracy, usability, and cost-effectiveness are largely absent from the literature. This gap highlights the need for comprehensive comparative analyses, such as the one presented in this study, that integrate academic insights with practical evaluations of AI-powered requirement prioritization tools.

Methodology

To ensure a structured and unbiased comparison of AI-powered requirement prioritization tools, this study employed a systematic, multi-dimensional evaluation framework. The methodology comprised three main components: tool selection criteria, evaluation dimensions, and data sources.

Tool Selection Criteria

The selection of tools was guided by inclusion criteria designed to ensure relevance, maturity, and verifiability. First, only tools under active development and support, with recent updates and accessible user communities, were considered. Second, tools were required to explicitly incorporate artificial intelligence (AI) or machine learning (ML) techniques such as natural language processing (NLP), sentiment analysis, or predictive modeling, as indicated in vendor documentation or technical descriptions. Third, compatibility with Agile software engineering environments was prioritized, with preference given to tools capable of integrating into widely used project management platforms such as Jira, Trello, GitHub, and Azure DevOps. Finally, transparency and accessibility were deemed essential, requiring publicly available documentation, pricing models, user reviews, and either trial versions or demos for empirical evaluation. These criteria reflect best practices for tool selection in requirements engineering research [10], [14], [17].

Evaluation Dimensions

The selected tools were assessed across four dimensions to provide a holistic comparison. The first dimension, AI capabilities, examined the depth and sophistication of AI integration, including NLP-based analysis, supervised learning models, and predictive prioritization mechanisms. The second dimension, cost and licensing, evaluated pricing structures such as subscription fees, enterprise licenses, freemium tiers, and potential hidden costs associated with onboarding or scaling. The third dimension, integration and usability, assessed the ability of each tool to embed within common Agile toolchains, its user interface quality, onboarding effort, and learning curve. Finally, advantages and limitations were considered, focusing on scalability, stakeholder satisfaction, use-case diversity, and reported constraints. This framework builds on prior evaluation models in requirements engineering and decision analysis [18], [19].

Data Sources

The analysis drew upon multiple sources to ensure methodological rigor and reduce bias. Official product documentation was reviewed to establish baseline capabilities and vendor claims. User experiences were incorporated by analyzing reviews and testimonials from platforms such as G2, Capterra, and TrustRadius. To ground the evaluation in academic research, relevant literature, white papers, and technical blogs were also consulted [11], [12], [14]. Where possible, tools were empirically explored using publicly available demos, sandbox environments, or trial versions. In addition, expert insights were obtained from Agile practitioners and software engineers with direct experience using these platforms, providing valuable context on usability and scalability. The triangulation of these diverse data sources enhanced the reliability and validity of the findings.

Comparative Analysis of AI Tools

This section presents a comparative analysis of selected AI-powered requirement prioritization tools. The objective is to examine their relative strengths and limitations across dimensions of AI capabilities, cost and licensing, integration and usability, and practical advantages and constraints. The tools analyzed include Airfocus, Craft.io, ClickUp AI, ReqSuite RM, Kanoah Tests, Zepel, and Aha! Roadmaps. These platforms were chosen on the basis of their active development, explicit integration of AI/ML techniques, and compatibility with Agile project management environments, as outlined in Section 4.

Table 1 provides a structured comparison of the tools. The analysis is based on product documentation, user reviews, academic literature, and expert practitioner insights, ensuring triangulation of data sources [10], [14], [17].

Table 1. AI-Powered Requirements Engineering (RE) Tools Comparison

Airfocus	ML-based scoring, Priority Poker	$$$ (Subscription)	Jira, Trello, Azure	★★★★☆	Mid-large teams	Custom prioritization, visual roadmaps	High cost for small teams
Craft.io	Predictive scoring, decision matrix	$$ (From $39/mo)	Azure, GitHub, Jira	★★★☆☆	Agile/SAFe teams	Supports WSJF/RICE frameworks	Complex setup
ClickUp AI	NLP-powered assistant	+$ (Add-on)	GitHub, Jira	★★★★☆	General automation	AI writing, task suggestions	Not RE-specific

ReqSuite RM	NLP-based auto-linking & tracing	$$$$ (Enterprise)	Jira, DOORS	★★☆☆☆	Enterprise teams	Strong automation, compliance support	Expensive, steep learning curve
Kanoah Tests	Rule-based automation	$$ (Subscription)	Jira	★★★☆☆	Test + RE integration	Combines RE and testing workflows	Limited AI sophistication
Zepel	Smart prioritization suggestions	$ (Flat-rate)	GitHub, Trello	★★★★☆	Small teams	Affordable, fast deployment	Narrow AI scope
Aha! Roadmaps	AI-driven prioritization scoring	$$$ (Professional)	Jira, Trello, GitHub	★★★☆☆	Product managers	AI-powered planning support	High cost, lacks ML transparency

Observations Across Tools

The comparative analysis reveals several important patterns.

AI Capabilities: Tools such as Airfocus and Craft.io employ predictive scoring and ML-based approaches that align with established prioritization frameworks like Weighted Shortest Job First (WSJF) [19]. ClickUp AI incorporates NLP to support general task interpretation, which resonates with recent studies on the application of NLP in requirements engineering [12], [15], [17]. In contrast, Kanoah Tests and Zepel provide more lightweight or rule-based features, which are accessible to smaller teams but lack advanced scalability.

Cost Models: Pricing strategies vary significantly, with enterprise-focused tools such as ReqSuite RM adopting high-cost licensing structures, while Zepel offers flat-rate affordability. These findings are consistent with earlier reports that cost is a critical barrier in the adoption of prioritization techniques [10]. The existence of freemium or add-on models, as seen in ClickUp AI, demonstrates attempts to lower entry barriers for smaller organizations.

Integration and Usability: Enterprise-grade tools such as ReqSuite RM provide strong traceability and compliance support by integrating deeply with platforms like Jira and DOORS, reflecting research emphasizing integration as a determinant of adoption [14]. Conversely, lightweight tools such as Zepel and ClickUp AI are easier to onboard but provide narrower functionality, which may constrain their use in larger or more regulated environments.

Advantages and Limitations: The tools collectively demonstrate trade-offs between cost, sophistication, and usability. Airfocus and Craft.io provide feature-rich environments but are often complex or costly for small organizations. ReqSuite RM excels in automation and compliance but has a steep learning curve, confirming prior concerns regarding adoption challenges in RE practices [10], [20]. Affordable solutions like Zepel demonstrate agility but lack advanced AI transparency, which is increasingly recognized as critical to user trust [13], [21].

Summary of Findings

The comparative analysis highlights that no single tool is universally optimal. Instead, suitability is shaped by organizational scale, budget, and maturity level. These findings align with prior literature on multi-criteria decision-making in RE, which emphasizes that trade-offs are context-dependent [18], [19]. Importantly, the lack of explainability features across most tools indicates an unresolved research gap, underscoring the need for XAI integration in future requirement prioritization platforms [13], [20], [21].

Discussion

The comparative analysis demonstrates that AI-powered requirement prioritization tools vary considerably in terms of capabilities, usability, and cost-effectiveness. This section interprets the results and situates them within existing research, focusing on three core themes: cost-benefit trade-offs, integration within Agile ecosystems, and human–AI collaboration.

Cost–Benefit Trade-offs

Cost remains one of the most critical differentiators among the tools. Platforms such as Zepel and ClickUp

AI offer entry-level affordability, providing accessible AI-driven features for small teams. In contrast, ReqSuite RM and Craft.io adopt enterprise-level licensing models, offering richer functionality at the expense of higher costs and training overheads. This finding is consistent with empirical evidence that financial and resource constraints are barriers to the adoption of requirements prioritization techniques in industry [10].

Moreover, the cost dimension aligns with broader research on decision-making frameworks, where trade-offs between functionality and affordability are context-dependent [18], [19]. For smaller organizations, the marginal benefits of advanced AI features may not justify the associated costs. For larger enterprises, however, advanced capabilities such as compliance, traceability, and predictive analytics may outweigh higher pricing.

Integration within Agile Ecosystems

Integration with existing Agile toolchains is another decisive factor influencing tool adoption. Tools such as Airfocus and ReqSuite RM provide deep integration with Jira, DOORS, and CI/CD pipelines, thereby supporting traceability and automation across the development lifecycle. Prior studies confirm that integration capabilities are a prerequisite for adoption, as isolated tools are often resisted by practitioners [14].

By contrast, lightweight tools such as Zepel and ClickUp AI prioritize rapid onboarding and ease of use, aligning with findings that usability is an important determinant of tool acceptance [20]. However, their limited integration capabilities may restrict their application in enterprise environments where compliance and scalability are critical.

Human–AI Collaboration and Explainability

A key concern emerging from the analysis is the opacity of AI-generated prioritization results. With the exception of limited rule-based approaches, most tools provide little transparency regarding how prioritization decisions are derived. This raises challenges for stakeholder trust, particularly in regulated industries where accountability is essential. The importance of explainability in requirements engineering has been emphasized in recent literature, which argues that the absence of transparency undermines stakeholder acceptance of AI recommendations [13], [20], [21].

Hybrid approaches, in which AI-generated recommendations are complemented by stakeholder validation, have been proposed as a means of addressing this gap [14]. Such models support the dual goals of efficiency and interpretability, aligning with Bano and Zowghi's [20] emphasis on the critical role of user involvement in ensuring system success. Future advancements in explainable AI (XAI) hold promise for bridging this trust gap, enabling wider adoption of AI-based prioritization tools in practice.

Implications for Research and Practice

The findings underscore that no tool is universally optimal; instead, suitability is contingent on organizational size, maturity level, and project context. For practitioners, the study provides evidence-based insights into the cost, usability, and integration trade-offs that must be considered when selecting tools. For researchers, the analysis highlights gaps in empirical validation, benchmarking, and explainability.

Specifically, future research should focus on longitudinal evaluations of tool performance in real-world projects, the development of open-source alternatives to democratize access, and the integration of XAI mechanisms to improve stakeholder trust. These directions are consistent with calls in the literature for hybrid, human–AI decision-making models that balance automation with contextual expertise [14], [17], [20].

AI Architecture in Requirement Prioritization

AI-powered requirement prioritization tools typically adopt a layered architecture that integrates natural language processing (NLP), machine learning (ML), and stakeholder feedback mechanisms. This architecture reflects common patterns in intelligent requirements engineering frameworks, where heterogeneous inputs are processed through sequential analytical stages to generate ranked backlogs [11], [12], [14], [15], [17].

Architectural Layers

Figure 1 illustrates the generic pipeline of AI-based prioritization systems, which consists of four primary layers: data ingestion, NLP-based preprocessing, ML-driven scoring, and prioritization output.

Data Collection:

The process begins with the aggregation of requirement artifacts, including user stories, feature requests, support tickets, and customer feedback. Research has emphasized the importance of diverse data sources to ensure comprehensive coverage of stakeholder needs [2], [12].

NLP Processing:
The collected data is preprocessed using NLP techniques such as tokenization, semantic similarity analysis, and ambiguity detection [12], [15], [21]. This step enables the system to interpret the context, intent, and relationships embedded within natural language requirements.
ML Scoring Engine: A machine learning model assigns priority weights based on predefined criteria, historical project data, or predictive analytics. Approaches such as supervised learning, sentiment analysis, and hybrid rule-based/ML methods are commonly used [14], [16], [17].
Priority Backlog Generation: The output layer produces a ranked list of requirements, often visualized in roadmaps or dashboards to facilitate Agile planning.

Feedback Loops and Adaptivity
A critical feature of modern AI-driven architectures is the integration of stakeholder feedback into iterative learning loops. Feedback may be collected through surveys, direct validation, or retrospective analysis of prior releases. This aligns with adaptive learning paradigms in RE, where systems improve continuously based on evolving stakeholder priorities [14], [20].

Challenges and Research Directions
Despite their promise, AI architectures for requirement prioritization face significant challenges. First, ambiguity and contradictions in natural language requirements remain difficult to resolve automatically [21]. Second, biases embedded in historical datasets can propagate into prioritization outcomes, raising concerns of fairness and accountability [13]. Third, most architectures lack built-in explainability mechanisms, reducing stakeholder trust in prioritization outputs [13], [20]. Addressing these challenges requires hybrid architectures that combine algorithmic predictions with human validation and transparent explainable AI (XAI) mechanisms [14], [17].

Conclusion
This study investigated the role of artificial intelligence (AI) in enhancing requirement prioritization within Agile software engineering. Through a structured comparative analysis of seven AI-powered tools—Airfocus, Craft.io, ClickUp AI, ReqSuite RM, Kanoah Tests, Zepel, and Aha! Roadmaps—the research highlighted trade-offs across AI capabilities, cost models, integration depth, and usability.
Findings indicate that AI has substantial potential to improve the scalability, objectivity, and efficiency of requirement prioritization compared to traditional manual methods [2], [11], [14]. However, no tool is universally optimal; suitability is contingent upon organizational scale, resource availability, and project context. Enterprise-oriented solutions such as ReqSuite RM offer advanced traceability and compliance support but entail high cost and steep learning curves, whereas lightweight tools like Zepel provide affordability and rapid onboarding but lack transparency and advanced AI features. These observations align with prior research emphasizing that multi-criteria decision-making in requirements engineering is inherently context-dependent [10], [18], [19].
Beyond cost and functionality, integration with Agile toolchains and the role of human–AI collaboration emerged as critical determinants of adoption. Tools with robust integration capabilities enhance traceability and workflow continuity, while the limited presence of explainable AI (XAI) mechanisms highlights a significant gap in stakeholder trust, particularly in regulated domains [13], [20], [21]. Overall, AI-driven requirement prioritization represents both a technological innovation and a strategic enabler of Agile success, with the potential to accelerate delivery, strengthen stakeholder alignment, and improve product outcomes.

Future Work and Improvements
While this study provides a comprehensive comparative analysis, several avenues remain for further exploration and refinement

Longitudinal Evaluation in Real-World Environments
Empirical assessments of AI-powered tools within operational Agile projects over extended periods are necessary to evaluate adaptability to evolving backlogs, stakeholder needs, and iterative delivery cycles.

Practitioner-Centered Insights
Structured surveys, interviews, and observational studies of Agile practitioners can illuminate usability challenges, adoption barriers, and trust factors, complementing technical evaluations with human-centered perspectives.

Inclusion of Open-Source Alternatives

The predominance of commercial tools limits accessibility for smaller or resource-constrained organizations. Future research should investigate open-source or community-driven frameworks to democratize access to intelligent prioritization capabilities.

Ethical and Bias Considerations

Bias, fairness, and data quality require further examination. Studies should assess how biased datasets or opaque AI models may affect prioritization outcomes, particularly in regulated or safety-critical domains, and explore mechanisms for bias mitigation.

Integration with Agile Practices

Research should examine the socio-technical impacts of AI tools on Agile team dynamics, sprint planning, and ceremonies, assessing whether AI enhances or disrupts collaboration.

Quantitative Benchmarking Frameworks

Developing standardized evaluation matrices and multi-criteria decision analysis models will enable objective comparisons of tools, providing measurable insights into trade-offs between cost, usability, AI sophistication, and organizational fit.

Emerging AI Techniques and Hybrid Intelligence Future studies should explore large language models (LLMs), reinforcement learning, and adaptive AI agents for requirement prioritization. Hybrid approaches that combine algorithmic recommendations with expert human judgment may optimize efficiency, explainability, and contextual relevance.

Addressing these directions will enhance the reliability, transparency, and practical adoption of AI-powered requirement prioritization tools, contributing to both academic knowledge and industrial practice.

REFERENCES

[1] B. Nuseibeh and S. Easterbrook, "Requirements Engineering: A Roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, 2000, pp. 35–46.

[2] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in Human Behavior*, vol. 51, pp. 915–929, Oct. 2015.

[3] Airfocus, "AI-Powered Product Management Software," 2025. [Online]. Available: https://airfocus.com

[4] Craft.io, "Product Management Platform," 2025. [Online]. Available: https://craft.io

[5] ClickUp, "ClickUp AI: Work Faster with AI Tools," 2025. [Online]. Available: https://clickup.com

[6] ReqSuite RM, "Requirements Management Software by OSSENO," 2025. [Online]. Available: https://osseno.com/requisite

[7] Kanoah Tests, "Test Management for Jira," 2025. [Online]. Available: https://marketplace.atlassian.com

[8] Aha! Roadmaps, "Product Roadmap Software," 2025. [Online]. Available: https://www.aha.io/roadmapping

[9] Zepel, "Project Management for Product Teams," 2025. [Online]. Available: https://zepel.io

[10] A. A. Khan, M. Niazi, and R. Ahmad, "Barriers in the adoption of software requirements prioritization techniques: An empirical investigation," *Information and Software Technology*, vol. 87, pp. 124–136, 2017. https://doi.org/10.1016/j.infsof.2017.02.004

[11] Z. Li, Z. Ma, and Q. Wang, "AI-based requirements engineering: A survey," in *28th IEEE International Requirements Engineering Conference (RE)*, 2020, pp. 364–369. https://doi.org/10.1109/RE48521.2020.00047

[12] A. Mahmoud and N. Niu, "A case for natural language processing in requirements engineering," in *27th IEEE International Requirements Engineering Conference (RE)*, 2019, pp. 17–27. https://doi.org/10.1109/RE.2019.00012

[13] H. Zhang, S. Yang, and T. Y. Chen, "Explainable artificial intelligence in requirements engineering: Opportunities and challenges," *ACM Computing Surveys*, vol. 51, no. 6, pp. 123:1–123:37, 2018. https://doi.org/10.1145/3274325

[14] J. Chen, M. A. Babar, and H. Zhang, "Towards intelligent requirements engineering: Leveraging machine learning techniques," *Journal of Systems and Software*, vol. 147, pp. 435–452, 2019. https://doi.org/10.1016/j.jss.2018.10.065

[15] P. Sawyer, P. Rayson, and A. G. C. P. Ward, “Improving requirements engineering with NLP: Results from industry,” in *IEEE 16th International Requirements Engineering Conference*, 2008, pp. 425–430. https://doi.org/10.1109/RE.2008.48

[16] E. Guzmán, D. Azócar, and Y. Li, “Sentiment analysis of commit comments in GitHub: An empirical study,” *Proceedings of the 11th Working Conference on Mining Software Repositories (MSR)*, pp. 352–355, 2014. https://doi.org/10.1145/2597073.2597118

[17] F. Dalpiaz, A. Ferrari, X. Franch, and P. Spoletini, “Natural language processing for requirements engineering: The best is yet to come,” *IEEE Software*, vol. 35, no. 5, pp. 115–119, 2018. https://doi.org/10.1109/MS.2018.3571222

[18] M. Glinz, “On non-functional requirements,” in *15th IEEE International Requirements Engineering Conference (RE)*, 2007, pp. 21–26. https://doi.org/10.1109/RE.2007.45

[19] P. Berander and P. Jönsson, “Hierarchical cumulative voting (HCV) prioritization: An experimental study of HCV and the pair-wise method,” in *International Symposium on Empirical Software Engineering*, 2006, pp. 17–26. https://doi.org/10.1109/ISESE.2006.261353

[20] S. Bano and D. Zowghi, “A systematic review on the relationship between user involvement and system success,” *Information and Software Technology*, vol. 58, pp. 148–169, 2015. https://doi.org/10.1016/j.infsof.2014.06.011

[21] A. Ferrari, P. Spoletini, and S. Gnesi, “Ambiguity detection in natural language requirements: An empirical study,” *Requirements Engineering*, vol. 22, no. 3, pp. 289–313, 2017. https://doi.org/10.1007/s00766-016-0255-8