

Course Title: Research Methodology in Computer Science

Course Code: AI800

Weighting: 30%

Lecturer: Dr. Mohammed Anbar

Hand-out Date: 1 November 2025

Hand-in Date: 25 November 2025

Hand-in Method: Online Submission via E-Learning.

Student Name: Nasser Abdullah Alshalwan

Student ID: 253000228

Assignment Report: Improving Software Quality in Agile Development Environments Using Artificial Intelligence Tools

Table of contents:

- Problem Statement
- Research Objectives
- Abstract
- Literature Review
- Summary Table
- Conclusion
- References

Literature Review: Improving Software Quality in Agile Development Environments Using Artificial Intelligence Tools

Problem Statement:

Agile software development has become a dominant approach for delivering software systems, especially in dynamic domains where requirements change frequently. While Agile methods such as Scrum and Kanban emphasize iterative delivery, close collaboration with stakeholders, and responsiveness to change, ensuring and maintaining high software quality remains a persistent challenge. Agile teams must balance short iterations, limited documentation, and rapid release cycles with the need for systematic testing, defect prevention, and continuous quality improvement. As projects grow in size and complexity, manual quality assurance activities alone become difficult to sustain and may lead to undetected defects, technical debt, or reduced customer satisfaction.

Artificial Intelligence (AI) tools offer promising opportunities to support software quality in Agile environments. Recent studies show that AI techniques—including machine learning, deep learning, and natural language processing—can assist Agile teams in activities such as test case generation, defect prediction, analysis of user stories, and automated code review (Cabrero-Daniel, 2023; Alenezi & Akour, 2025; Ahmed et al., 2023). Other research explores how Generative AI and predictive analytics can help prioritize testing and improve decision-making in Agile projects (Cornide-Reyes et al., 2025; Christadoss et al., 2025). Despite these advances, the adoption of AI in Agile practice is still emerging and uneven.

The reviewed literature highlights several limitations and open issues. Many works are conceptual or based on small case studies, with limited large-scale empirical evaluation across diverse Agile teams (Peras et al., 2023; Christadoss et al., 2025). Some proposed tools are not specifically tailored to Agile workflows, which makes integration into everyday practices challenging. Data quality, language diversity, and domain-specific terminology can reduce the effectiveness of NLP-based techniques for user stories (Ahmed et al., 2023). In addition, there are concerns about the transparency and explainability of AI models, which may affect trust and acceptance by developers and stakeholders. Therefore, there is a need for a structured synthesis of existing research to clarify how AI tools can realistically improve software quality in Agile environments, what benefits have been demonstrated, and which gaps must be addressed in future work.

Research Objectives:

Primary Objective

To synthesize and evaluate current research on the use of Artificial Intelligence tools to improve software quality in Agile development environments, and to identify key benefits, limitations, and directions for future work.

Supporting Objectives:

1. **To analyze how AI techniques are applied to support quality-related activities in Agile projects**, such as defect prediction, test generation, code analysis, and user story evaluation.
2. **To compare the methods and evidence reported in recent studies**, including research designs, datasets, evaluation metrics, and Agile contexts (e.g., Scrum, Kanban, or hybrid approaches).
3. **To identify the main strengths and weaknesses of AI-based approaches for Agile software quality**, focusing on issues such as data quality, tool integration, scalability, and ease of use for development teams.
4. **To examine empirical limitations in the existing literature**, including small sample sizes, lack of long-term studies, and restricted industrial validation.
5. **To propose future research directions for applying AI in Agile software quality assurance**, emphasizing more robust empirical evaluations, better integration with Agile practices, and improved transparency and explainability of AI tools.

Abstract:

This paper reviews how Artificial Intelligence (AI) tools can be used to improve software quality in Agile development environments. The focus is on AI techniques such as machine learning models (e.g., Random Forest, XGBoost), neural networks, and natural language processing (NLP) that support activities like test generation, defect prediction, and analysis of user stories. The review is based on six recent studies that examine AI support for Agile planning, coding, debugging, and quality assurance. Overall, the literature shows that AI can help teams detect defects earlier, automate repetitive quality tasks, and save time. However, the studies also report several challenges, including limited real-world evaluation, data quality issues, lack of long-term evidence, and difficulties in making AI-based tools explainable and easy to adopt in practice.

Literature Review:

Recent research has explored how AI can support software quality in Agile projects from different perspectives. Cabrero-Daniel (2023) presents *AI for Agile Development*, a meta-analysis that describes how AI techniques assist Agile activities such as planning, estimation, and testing. The study provides a broad overview and highlights many possible uses of AI, but it includes limited empirical data to quantify the actual quality improvements.

In *AI-Driven Innovations in Software Engineering*, Alenezi and Akour (2025) discuss how AI can improve code quality and debugging. Their review focuses on tools that automatically detect defects or suggest fixes. The paper offers strong, concrete examples, but not all of the discussed techniques are tailored specifically to Agile processes, which makes it harder to generalize the findings to Scrum or Kanban teams.

Generative AI has also been examined as a way to support Agile development. Cornide-Reyes et al. (2025) analyze how GenAI models can help during planning, coding, and testing in Agile projects. They describe phases where GenAI can generate code, tests, or documentation. While the paper explains these phases clearly, it does not provide long-term empirical results on project outcomes or team productivity.

Other studies focus on cataloguing AI techniques used in Agile environments. Peras et al. (2023) present *AI Techniques and Tools in Agile Software Development*, which maps different AI tools to Agile practices. The mapping is well organized and useful as a reference, but the work reports only a few experimental evaluations of the listed tools.

Textual artefacts in Agile, such as user stories, have also been analyzed using NLP. Ahmed et al. (2023) propose NLP-based techniques to assess the quality of user stories. Their work is tested in real projects and shows that NLP can detect ambiguous or incomplete stories, which may reduce defects later. However, the study notes that language and domain issues can limit accuracy when applying the same approach to other teams or languages.

Finally, Christadoss et al. (2025) investigate *Predictive Quality Analytics Using ML* in Agile settings. They use machine learning models, including XGBoost, to predict defect-prone components and prioritize testing. The results indicate good predictive accuracy and potential time savings, but the evaluation is based on data from a single company, which restricts the generalizability of the conclusions.

Summary Table:

Year	Title	Method	Weakness
2023	AI for Agile Development	Review	Few quantitative results
2025	AI-Driven Innovations	Review	Not always focused on Agile
2025	Generative AI in Agile	Review	No long-term evaluation
2023	AI Techniques in Agile	Mapping study	Few experimental tests
2023	NLP for User Stories	Case study	Language and domain issues
2025	Predictive Quality with ML	Case study	Data from only one company

Conclusion:

The reviewed studies indicate that AI tools can significantly support software quality activities in Agile development, especially in defect prediction, automated testing, and analysis of user stories. Machine learning and NLP techniques help teams save time and identify quality issues earlier in the development cycle. At the same time, the literature highlights several limitations: many AI approaches are evaluated on small or single-company datasets, some tools are not explicitly designed for Agile workflows, and there is a lack of long-term evidence about their impact on project outcomes and team practices. Future research should therefore focus on conducting larger empirical studies in diverse Agile environments, improving the explainability of AI-based tools, and integrating them more seamlessly into everyday Agile processes.

References:

1. Cabrero-Daniel, B. (2023). *AI for Agile Development: A Meta-Analysis*.
2. Alenezi, M., & Akour, M. (2025). *AI-Driven Innovations in Software Engineering*.
3. Cornide-Reyes, H., et al. (2025). *Generative AI in Agile Software Development*.
4. Peras, D., et al. (2023). *AI Techniques and Tools in Agile Software Development*.
5. Ahmed, M., et al. (2023). *NLP-Based Quality Attributes Extraction*.
6. Christadoss, J., et al. (2025). *Predictive Quality Analytics Using ML*.