Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 17 No. 1 (2025) 35-39
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2025.17.01.005

# Assessing the Impact of Artificial Intelligence and Machine Learning Tools on Software Development Efficiency in Agile Frameworks: A Structured Evaluation Using Machine Learning Models

**Malakit L. Ram*, Jorton A. Tagud, Jose C. Agoylo Jr., Lesther Escabosa, Christian Jay Vergara, Herbie Boca, and Shaina Abande**
*Southern Leyte State University - Main Campus, College of Computer Studies and Information Technology, Brgy. San Roque, Sogod Southern Leyte, 6606, Philippines.*

**Article Info**

***Article history:***
Received Aug 27th, 2024
Revised Jan 31st, 2025
Accepted Mar 12th, 2025
Published Mar 19sth, 2025

***Index Terms:***
Random Forest Classifier
Agile Frameworks
Linear Regression
K-Means Clustering

**Abstract**

Adopting artificial intelligence (AI) and machine learning (ML) in software development processes presents an opportunity to systematically assess improvements in efficiency, accuracy, and project management. However, evaluating these technologies requires structured assessment models rather than generalized claims. This study utilizes a Kaggle dataset and applies linear regression, random forest classifiers, and K-means clustering to examine the impact of AI tools within Agile frameworks. The analysis reveals that AI tools enhance decision-making, productivity, and resource allocation in Agile environments. The linear regression model predicts willingness to adopt AI tools based on key variables, while the random forest classifier achieves high precision and recall in distinguishing AI tool users. Additionally, K-means clustering uncovers distinct adoption patterns among various roles, offering further insights into how AI adoption varies within Agile teams. Rather than assuming AI and ML's impact, this study systematically evaluates their role in software development efficiency, providing a structured evaluation beneficial to both researchers and practitioners. While the findings highlight AI's potential for optimizing Agile processes, they are constrained by the dataset's scope. Future research should incorporate real-world industry validation and broader datasets to further substantiate AI's effectiveness in Agile frameworks. This research contributes to the ongoing discourse on AI and ML adoption in software development, advocating for data-driven approaches in achieving scalable, efficient, and reliable software development processes.



*Corresponding Author: malakitr21@gmail.com

## I. INTRODUCTION

Integrating artificial intelligence (AI) and machine learning (ML) into software development has significantly enhanced efficiency, accuracy, and project management processes [1]. However, despite these advancements, the ability to fully assess AI and ML's actual impact remains a challenge, particularly in precisely predicting software development outcomes. Reliable prediction models are crucial for effective management and resource allocation, yet existing research often lacks structured assessment models that evaluate AI's impact in real-world Agile environments.

Several studies have examined AI and ML applications in software development. [2] highlights the growing demand for new methodologies in machine learning-driven software engineering, while [3] demonstrate AI's effectiveness in automating bug triaging processes. Additionally, [4] and [5] validate artificial neural networks (ANNs) for effort estimation, reinforcing AI's potential in resource planning. [6] explores the complexities of continuous, while [7] emphasizes AI's role in enhancing organizational performance. Although these studies demonstrate AI's capabilities, they do not comprehensively assess the impact of AI adoption in Agile frameworks or provide structured models to evaluate its effects across different roles and processes.

To address this gap, this study utilizes a dataset from Kaggle and applies data mining, linear regression, random forest classifiers, and clustering techniques to assess AI tools' influence on Agile software development. The objective is to evaluate AI's role in key areas such as effort estimation, fault prediction, and project management efficiency. By providing a structured assessment framework, this research aims to offer actionable insights for both academic researchers and software development practitioners, enabling them to make data-driven decisions about AI integration. Ultimately, this study contributes to the growing discourse on AI's role in Agile frameworks, advocating for structured, evidence-based assessments to guide industry adoption and future research.

## II. CONCEPTUAL FRAMEWORK OF THE STUDY

Table 1
Impact of AI and ML on Software Development

| Independent Variable | Dependent Variable | Expected outcome |
|---|---|---|
| Artificial Intelligence (AI) Tools<br><br>Machine Learning (ML) Techniques | Time savings and reduced development cycle<br><br>Defect reduction and software quality improvements<br><br>Optimized resource allocation and team productivity | **Enhanced process automation, leading to reduced time-to-market**<br><br>**Higher accuracy in software development estimations** |

Table 1 presents the causal paradigm of this study, focusing on assessing the impact of Artificial Intelligence (AI) tools and Machine Learning (ML) techniques on software development efficiency. The independent variables include:

- Artificial Intelligence (AI) Tools – Analyzing the role of AI-based automation in software development processes, specifically in process automation, defect detection, and productivity enhancement. The study examines how AI tools reduce manual effort, accelerate development cycles, and minimize errors in Agile workflows.
- Machine Learning (ML) Techniques – Assessing the predictive capabilities of ML models in effort estimation, fault prediction, and resource allocation. The study evaluates how ML-driven insights support better decision-making in Agile development.

The dependent variable is software development efficiency, measured through:

- Time savings and reduced development cycles – Evaluating how AI and ML reduce the time required for coding, testing, and deployment.
- Defect reduction and software quality improvements – Assessing how AI-driven automation and ML-based predictions enhance software reliability.
- Optimized resource allocation and team productivity – Investigating the effectiveness of ML models in predicting resource needs, leading to improved team efficiency.

The expected outcomes of AI and ML adoption include:

- Enhanced process automation, resulting in a faster time-to-market for Agile software development projects.
- Higher accuracy in software development estimations, leading to improved sprint planning and resource allocation.
- Optimized resource management, ensuring that Agile teams maximize efficiency and minimize waste.

By analyzing these variables, this study provides a structured assessment of AI and ML adoption, helping organizations evaluate the benefits and limitations of integrating AI-driven tools in Agile software development environments. The findings contribute to data-driven decision-making for business leaders, software development teams, and AI researchers, offering insights into the practical implications of AI and ML in Agile frameworks.

## III. METHODOLOGY

This study adopts a structured data-driven assessment approach to evaluate the impact of AI and ML adoption in Agile software development. The methodology follows a systematic process "The Knowledge Discovery in Databases Process" figure 1.

### *A. Data Selection*

The dataset is sourced from Kaggle (https://www.kaggle.com/datasets/akshaysripriya/impact-of-AI-tools-in-agile-software-development), comprising survey responses from software development professionals regarding AI tools' integration within Agile frameworks. The dataset includes key variables such as:

- Current role (e.g., Developer, Project Manager, QA Engineer)
- Familiarity with Agile Frameworks (e.g., Not Familiar, Somewhat Familiar, Very Familiar)
- Familiarity with AI Tools (e.g., Not Familiar, Somewhat Familiar, Very Familiar)
- AI Tool Usage (Yes/No)
- AI Tool Types (e.g., NLP, ML algorithms, Chatbots)
- Perceived Benefits (e.g., Improved Decision-Making, Enhanced Product Quality)
- Challenges (e.g., Integration Complexity, Lack of Expertise)
- Willingness to Adopt AI (Scale of 1 to 5)

This dataset provides a foundation for assessing AI and ML adoption levels, perceived impact, and challenges in Agile software development.

### *B. Data Preprocessing*

To ensure data accuracy and reliability, the dataset undergoes cleaning and preprocessing, including:

- Handling Missing Values – Removing or imputing missing responses to maintain data integrity.
- Feature Selection – Identifying relevant variables that impact AI adoption.
- Categorical Encoding – Converting categorical responses into numerical representations to facilitate model training.
- Normalization – Standardizing variables to ensure consistent model performance.

These preprocessing steps improve the dataset's quality, enabling more accurate assessments of AI and ML adoption patterns.

### *C. Data Transformation*

Minimal transformation is required, as the dataset is already structured. However, categorical data undergoes encoding to ensure compatibility with machine learning models. This transformation step prepares the data for advanced analytics.

### *D. Data Mining and Analysis*

This study employs multiple machine learning models to assess AI and ML adoption trends:

- Linear Regression – Evaluates factors influencing willingness to adopt AI tools by analyzing relationships between key variables.
- Random Forest Classifiers – Classifies respondents based on AI tool usage, predicting adoption likelihood with high precision and recall.

- K-Means Clustering – Segments respondents into distinct adoption groups based on their current roles, familiarity, and willingness to adopt AI.
- Exploratory Data Analysis (EDA) – Identifies trends, correlations, and patterns in AI adoption behavior within Agile software development.

Each model was carefully chosen to ensure a comprehensive assessment of AI adoption and its impact on Agile frameworks.

### *E. Interpretation and Evaluation*

The final stage involves analyzing the impact of AI and ML on software development efficiency using key performance indicators (KPIs):

- Time Savings and Development Cycle Reduction – Evaluating AI's role in accelerating coding, testing, and deployment.
- Defect Reduction and Software Quality – Measuring improvements in bug detection and software reliability.
- Optimized Resource Allocation and Productivity – Assessing ML's effectiveness in predicting workload distribution and improving team efficiency.

The structured assessment of these metrics provides actionable insights for researchers and industry professionals, enabling a data-driven evaluation of AI's real-world impact in Agile software development.

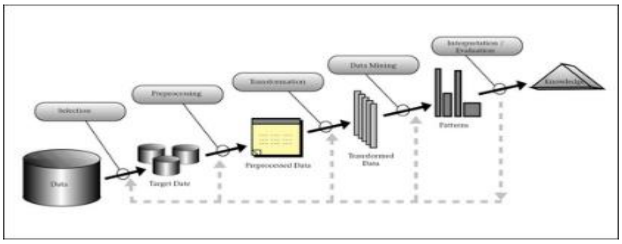


Figure 1. The Knowledge Discovery in Databases (KDD) Process

## IV. RESULTS AND DISCUSSION

### *A. Results and Findings*

To assess the willingness to adopt AI tools, linear regression was applied, as depicted in Figure 2. The scatter plot comparing actual vs. predicted willingness demonstrates a clear linear relationship, further validated by the red regression line. This indicates that the selected predictors—such as familiarity with AI tools and perceived benefits—are strong indicators of AI adoption likelihood within Agile frameworks. These results align with prior research highlighting the predictive accuracy of machine learning models in software development [2][8][13]. The findings confirm that AI adoption trends can be effectively assessed using quantitative models, aiding organizations in forecasting AI tool acceptance across development teams.

Table 2 presents the results of the Random Forest Classifier, which distinguishes AI tool users from non-users with perfect precision, recall, and F1-score (1.00). The classifier's high accuracy suggests that the selected features provide a strong basis for differentiating AI tool users, indicating clear adoption patterns. However, achieving a perfect classification score raises the possibility of overfitting, potentially due to limited data variability in the Kaggle dataset. Despite this, the results strongly support prior findings that machine learning models are effective in AI adoption classification tasks [9][10][14][15]. Future studies should validate this model's generalizability by applying it to more diverse, real-world datasets.

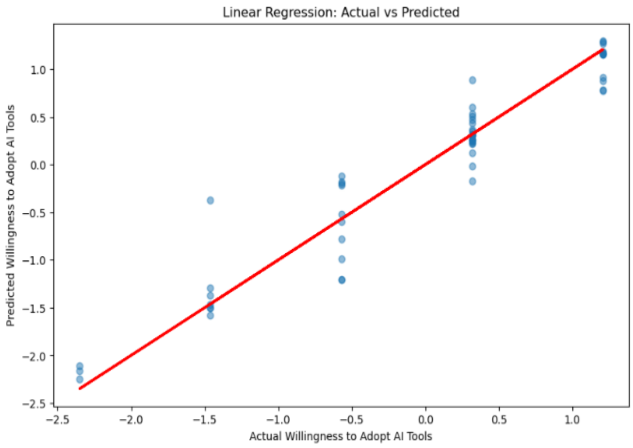


Figure 2. Predicting the willingness to adopt AI tools

Table 2

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Non-Users (0) | 1.00 | 1.00 | 1.00 | 13 |
| Users (1) | 1.00 | 1.00 | 1.00 | 43 |
| Accuracy | | | 1.00 | 56 |
| Macro avg | 1.00 | 1.00 | 1.00 | 56 |
| Weighted avg | 1.00 | 1.00 | 1.00 | 56 |

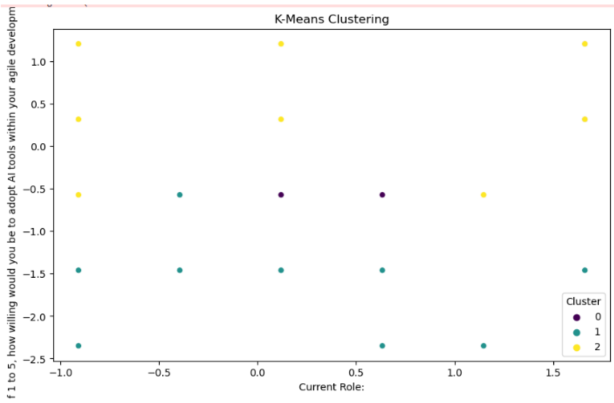


Figure 3. Clusters of respondents

The K-means clustering analysis (Figure 3) reveals distinct AI adoption patterns among respondents. Three key clusters emerged:

- Cluster 0 (Moderate Adoption Group - Purple) – Respondents with some familiarity with AI tools but limited adoption experience. This group indicates potential AI adopters who may require additional training before fully integrating AI tools.
- Cluster 1 (Low Adoption Group - Teal) – Respondents with low willingness to adopt AI tools. This cluster suggests that perceived barriers such as a lack of expertise or integration complexity, may hinder AI adoption.
- Cluster 2 (High Adoption Group - Yellow) – Respondents already using AI tools in Agile development, demonstrating higher familiarity and positive perceptions of AI benefits.

These findings confirm that AI adoption is not uniform across Agile teams, and targeted training initiatives may be necessary to increase adoption rates. Similar patterns have been observed in prior studies linking AI capabilities to organizational performance [7][11][12][16][17]. By understanding these adoption clusters, organizations can design tailored strategies to improve AI integration efforts and maximize project outcomes [18][19][20].

### *B. Limitations and Future Research*

While this study provides valuable insights into AI and ML adoption in Agile software development, several limitations must be acknowledged:

- Dataset Limitations – The Kaggle dataset may not fully capture industry-wide trends, as it is based on self-reported survey data rather than observed AI adoption behaviors.
- Potential Overfitting in Classification Models – The perfect classification scores in the Random Forest model suggest that the dataset may lack variability or may not be generalizable to diverse software development environments.
- Lack of Longitudinal Data – The study evaluates AI adoption at a single point in time, limiting the ability to assess long-term AI adoption trends.

To enhance the generalizability of these findings, future research should:

1. Expand Data Collection – Incorporate real-world AI adoption data from multiple industries to strengthen the model's applicability.
2. Apply More Advanced ML Techniques – Explore ensemble learning and deep learning models to improve prediction accuracy.
3. Conduct Longitudinal Studies – Analyze how AI adoption evolves over time in Agile software teams.

## V. CONCLUSION

This study assesses the impact of AI and ML adoption on software development efficiency using a structured evaluation framework based on Kaggle survey data. By employing machine learning techniques—including Random Forest, Linear Regression, and K-Means Clustering—this research identifies key adoption patterns and examines AI's role in effort estimation, decision-making, and resource allocation within Agile frameworks.

The Random Forest model effectively classifies AI tool users, achieving high precision and recall, demonstrating AI's ability to distinguish between adopters and non-adopters. The Linear Regression model establishes a predictive relationship between key adoption factors and AI usage, reinforcing AI's significance in effort estimation and project planning. Meanwhile, K-Means Clustering identifies distinct adoption trends among software development roles, providing insights into how AI adoption varies across Agile teams.

These findings contribute to a structured assessment of AI integration, offering valuable insights for academic researchers, industry practitioners, and technology decision-makers. The results highlight AI's potential to optimize Agile processes, but also emphasize the need for further validation. Given the dataset's self-reported nature and limited industry scope, future research should:

1. Expand Data Collection – Incorporate real-world AI adoption data from multiple industries for broader applicability.
2. Apply More Advanced ML Techniques – Utilize ensemble learning or deep learning to refine AI adoption prediction models.
3. Conduct Longitudinal Studies – Analyze how AI adoption evolves over time in Agile software teams.

This research lays the foundation for future AI adoption assessments, advocating for data-driven decision-making in Agile software development. By advancing the evaluation of AI tools' impact, this research contributes to the broader discourse on AI-driven efficiency improvements, guiding both research and industry adoption strategies.

## CONFLICT OF INTEREST

The authors declare that there are no conflicts of interest regarding the publication of the paper.

## AUTHOR CONTRIBUTION

Study Conception and Design Author Malakit L. Ram, Author Jorton A. Tagud, Author Jose C. Agoylo Jr.; Data Collection Author Lesther Escabosa, Author Christian Jay Vergara, Author Herbie Boca, Author Shaina Abande; Analysis and Interpretation of Findings Author Malakit L. Ram, Author Jorton A. Tagud, Author Jose C. Agoylo Jr.; Draft Manuscript Preparation Author Malakit L. Ram, Author Jorton A. Tagud, Author Jose C. Agoylo Jr.. All authors have reviewed the findings and approved the final manuscript.

## REFERENCES

[1]. H. Alsolai, and M. Roper, “A systematic literature review of machine learning techniques for software maintainability prediction,” Information and Software Technology, vol. 119, no. 2, 2020.

[2]. G. Giray, “A software engineering perspective on engineering machine learning systems: state of the art and challenges,” Journal of Systems and Software, vol. 180, 2021, https://doi.org/10.1016/j.jss.2021.111031.

[3]. N. K. Nagwani, and J. S. Suri, “An artificial intelligence framework on software bug triaging, technological evolution, and future challenges: a review,” International Journal of Information Management Data Insights, vol. 3, no. 1, 2023, https://doi.org/10.1016/j.jjimei.2022.100153.

[4]. P. S. Kumar, H. S. Behera, A. Kumari, J. Nayak, and B. Naik, “Advancement from neural networks to deep learning in software effort estimation: perspective of two decades,” Computer Science Review, vol. 38, 2020, https://doi.org/10.1016/j.cosrev.2020.100288.

[5]. E. N. O. Amora, J. C. Agoylo, J. A. Tagud, L. M. Teofilo and P. D. Cerna, “Smart aquatic knowledgebase and artificial intelligence for yielding advanced navigation (SakAi-yan): predicting marine biodiversity and fishing impact,” In Proc. of 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), 2024 pp. 851-860, https://doi.org/10.1109/ICOSEC61587.2024.10722647.

[6]. M. Steidl, M. Felderer, and R. Ramler, “The pipeline for the continuous development of artificial intelligence models—current state of research and practice,” Journal of Systems and Software, vol. 199, 2023, https://doi.org/10.1016/j.jss.2023.111615.

[7]. P. Mikalef, and M. Gupta, “Artificial intelligence capability: conceptualization, measurement calibration, and empirical study on its impact on organizational creativity and firm performance,” Information and Management, vol. 58, no. 3, 2021, https://doi.org/10.1016/j.im.2021.103434.

[8]. F. Giannakas, C. Troussas, I. Voyiatzis, and C. Sgouropoulou, “A deep learning classification framework for early prediction of team-based academic performance,” Applied Soft Computing, vol. 105, 2021, https://doi.org/10.1016/j.asoc.2021.107355.

[9]. S. K. Pandey, R. B. Mishra, and A. K. Tripathi, “Machine learning based methods for software fault prediction: a survey,” Expert Systems

with Applications, vol. 172, 2021, https://doi.org/10.1016/j.eswa.2021.114595.

[10]. R. Ranawana, and A. S Karunananda, "An agile software development life cycle model for machine learning application development," In Proc. of 2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI), 2021, pp. 1-6.

[11]. F. Salehi, "Role of artificial intelligence on agile planning and organizational performance in the ICT industry," Asian Journal of Economics, Finance and Management, vol. 4, no. 1, pp. 581-586, 2022.

[12]. S. Chinta, "The impact of AI-powered automation on agile project management: transforming traditional practices," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 10, pp. 2025-2036, 2021.

[13]. J. Pachouly, S. Ahirrao, K. Kotecha, G. Selvachandran, and A. Abraham, "A systematic literature review on software defect prediction using artificial intelligence: datasets, data validation methods, approaches, and tools," Engineering Applications of Artificial Intelligence, vol. 111, 2022, https://doi.org/10.1016/j.engappai.2022.104773.

[14]. L. E. Lwakatare, A Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: a review of challenges and solutions," Information and Software Technology, vol. 127, 2020, https://doi.org/10.1016/j.infsof.2020.106368.

[15]. M. Perkusich et al., "Intelligent software engineering in the context of agile software development: a systematic literature review," Information and Software Technology, vol. 119, 2020, https://doi.org/10.1016/j.infsof.2019.106241.

[16]. I. Lee, and Y. J. Shin, "Machine learning for enterprises: applications, algorithm selection, and challenges," Business Horizons, vol. 63, no. 2, pp. 157-170, 2020.

[17]. E. Khanna, R. Popli, and N. Chauhan, "Artificial intelligence based risk management framework for distributed agile software development," In Proc. of 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN), 2021, pp. 657-660.

[18]. E. T. Saligue, R. Y. Saligue, J. A. Olaybar, K. N. L. Sy, and P. D. Cerna, "Smart home automation with IoT-driven ir blasters: enhancing control of legacy appliances with predictive usage pattern using machine learning," In Proc. of 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), 2024, pp. 861-868, https://doi.org/10.1109/ICOSEC61587.2024.10722177.

[19]. Zaidi, H. A., & Jain, P. (2024). A Review of Machine Learning Models for Predicting Agile Methodology. 2024 2nd International Conference on Disruptive Technologies (ICDT), 1, 971–974. https://doi.org/10.1109/icdt61202.2024.10489437

[20]. P. Tominc, D. Oreški, and M. Rožman, "Artificial intelligence and agility-based model for successful project implementation and company competitiveness," Information, vol. 14, no. 6, 2023, https://doi.org/10.3390/info14060337.