Analyzing the Effectiveness of Agile Management Practices Through Random Forest and XGBoost-Based Performance Modeling

Quba Siddique[1,*]

[1]Institute of Banking and Finance, Bahauddin Zakariya University Multan, Pakistan

ABSTRACT

Agile management has become a central organizational strategy for coordinating software development teams in rapidly changing environments. However, the effectiveness of Agile practices is often evaluated subjectively, relying on qualitative assessments rather than data-driven evidence. This study investigates the managerial determinants of Agile team performance by applying Random Forest and XGBoost models to an AI-driven developer dataset. The analysis identifies the managerial factors that exert the strongest influence on task completion outcomes, focusing on resource allocation, risk assessment, project assignment, planning accuracy, and AI-enabled optimization. Results show that real-time resource prediction, AI optimization effectiveness, project overrun percentage, and risk assessment score are the most influential predictors of Agile performance, demonstrating that effective Agile Management depends on adaptive planning, proactive risk control, and strategic workload alignment. The study contributes to Agile literature by offering an empirical, model-based framework for evaluating managerial effectiveness and highlights the role of machine learning as a decision-support tool for enhancing Agile performance. These findings provide actionable insights for organizations seeking to strengthen Agile governance through predictive analytics and evidence-based management.

Keywords Agile Management, Machine Learning, Random Forest, XGBoost, Managerial Predictors

Submitted: 5 October 2024
Accepted: 10 December 2024
Published: 3 May 2025

Corresponding author
Quba Siddique,
qubassindhu@gmail.com

Additional Information and Declarations can be found on page 105

INTRODUCTION

Agile management has become a dominant paradigm in modern software development due to its capacity to support adaptability, collaboration, and rapid response to changing requirements [1]. As organizations increasingly operate in volatile and innovation-driven environments, Agile methodologies such as Scrum and Kanban enable iterative delivery, continuous feedback, and team autonomy, contrasting sharply with traditional predictive project management models [2]. These characteristics position Agile as a method capable of enhancing organizational responsiveness and reducing project uncertainty [3].

Despite its widespread adoption, evaluating the effectiveness of Agile management practices remains challenging. Many organizations rely on subjective assessments, qualitative observations, or retrospective meetings to determine whether managerial decisions—such as sprint planning, workload allocation, or risk mitigation—meaningfully contribute to performance outcomes [4]. Such reliance on qualitative judgment often obscures underlying behavioral and operational relationships, limiting managerial insight into which actions genuinely drive Agile team performance [5].

Recent advancements in data analytics and AI-driven development

How to cite this article: Q. Siddique, "Analyzing the Effectiveness of Agile Management Practices Through Random Forest and XGBoost-Based Performance Modeling," *Agile Manag.*, vol. 1, no. 2, pp. 92-107, 2025.

environments offer opportunities to augment Agile evaluation with quantitative evidence. Agile workflows increasingly generate detailed operational data, including resource consumption, task performance indicators, cycle times, risk scores, and real-time predictions produced by AI-assisted tools [6]. Machine learning techniques, particularly ensemble-based models such as Random Forest and XGBoost, have proven effective for identifying complex, non-linear relationships within such datasets. These models not only provide predictive capabilities but also reveal interpretable feature importance rankings that help determine which managerial factors exert the greatest influence on team performance [7].

However, existing research on Agile primarily focuses on adoption challenges, cultural and behavioral aspects, or team dynamics, while giving limited attention to data-driven managerial evaluation [8]. Most prior studies examine Agile implementation through qualitative or conceptual approaches rather than modeling how specific managerial decisions shape measurable performance outcomes [9]. This gap highlights the need for empirical methods that can quantify the effectiveness of Agile management in modern, data-rich development ecosystems.

The present study addresses this gap by applying Random Forest and XGBoost models to an AI-driven software development dataset to evaluate the influence of key managerial variables—including project assignment, risk assessment score, real-time resource prediction, developer experience, AI optimization effectiveness, and project overrun percentage—on task completion outcomes. The results show that Agile performance is strongly shaped by predictive planning accuracy, effective workload alignment, proactive risk mitigation, and integration of AI-enabled decision support, demonstrating that Agile Management effectiveness is fundamentally dependent on data-informed managerial practices.

The contributions of this research are threefold. First, it introduces a machine-learning-based framework for quantitatively assessing the effectiveness of Agile management using real performance data. Second, it provides empirical insights into the managerial determinants of Agile success by identifying the most influential predictors of performance. Third, it highlights the emerging role of AI and predictive analytics as decision-support mechanisms that enhance Agile governance and strengthen organizational capacity for adaptive planning.

Literature Review

Agile Management and Organizational Adaptability

Agile management has emerged as a foundational framework for software development teams operating in dynamic environments characterized by rapid technological evolution and shifting customer expectations [10]. Agile methodologies such as Scrum and Kanban emphasize iterative cycles, continuous feedback, and cross-functional collaboration, enabling organizations to respond more effectively to uncertainty and changing requirements [11]. These approaches have been shown to reduce project risk, enhance adaptability, and improve customer alignment compared to traditional plan-driven approaches [12].

However, the effectiveness of Agile methodologies depends heavily on how managers interpret and operationalize Agile principles within their teams. Poor sprint planning, unclear task distribution, and weak managerial oversight can undermine Agile adoption, even when teams formally implement its ceremonies and artifacts [13]. This indicates that Agile success is closely linked to managerial competence and decision-making rather than methodological compliance alone.

Managerial Determinants of Agile Team Performance

Managerial behaviors play a central role in shaping Agile team performance. Studies show that resource allocation strategy, workload balance, and real-time decision-making directly affect task completion rates and cycle-time efficiency [14]. Effective risk assessment is also essential, as unmanaged risks can disrupt workflow continuity and lead to delays that accumulate across sprint cycles [15]. Competency-based task assignment—matching tasks with developers' skills and experience—has been identified as a critical predictor of team-level productivity and code quality [16].

Additionally, project overrun metrics have been widely recognized as indicators of managerial planning accuracy. High overrun percentages often reflect weaknesses in estimation, inadequate backlog refinement, or misalignment between planned and actual team capacity [17]. These insights collectively emphasize the importance of systematic managerial evaluation within Agile environments.

AI-Enhanced Agile Environments

The integration of artificial intelligence has reshaped modern Agile development environments, offering new capabilities for prediction, automation, and decision support. AI-driven tools now assist developers and managers in workload estimation, anomaly detection, code generation, and risk forecasting, enhancing overall agility and workflow continuity [18]. Research indicates that effective AI adoption can reduce cognitive load, improve estimation accuracy, and support sprint stability through real-time analytics [19].

Metrics such as AI Optimization Effectiveness capture the extent to which teams utilize AI capabilities to enhance performance. Teams that effectively integrate AI tools tend to exhibit higher productivity and more consistent delivery outcomes, suggesting that AI adoption is not merely a technical enhancement but a managerial competency in modern Agile practice [20].

Machine Learning for Performance Evaluation in Agile Contexts

Machine learning has become an increasingly valuable tool for evaluating software engineering processes due to its ability to detect complex, non-linear patterns within operational data. Ensemble models such as Random Forest and XGBoost have been particularly effective in predicting project risks, developer workload, defect probabilities, and delivery outcomes [21]. Random Forest offers high interpretability through feature importance analysis, enabling researchers to identify which managerial factors most strongly influence performance [22]. XGBoost, known for its gradient-boosting optimization, provides high predictive accuracy and handles complex interactions across many variables [23].

Despite these advancements, few studies have applied machine learning explicitly to evaluate Agile management effectiveness. Much of the existing research focuses on behavioral or cultural dimensions of Agile adoption rather than quantifying which managerial decisions contribute most to performance outcomes [24]. This creates a gap in the literature where data-driven managerial evaluation is still underdeveloped.

Research Gap

A clear gap persists in the literature concerning the empirical, data-driven evaluation of Agile Management effectiveness. Prior research has largely emphasized subjective assessments, interviews, or survey-based evaluations of Agile maturity, offering limited insights into measurable managerial determinants of performance. Few studies leverage machine learning to identify how specific managerial variables—such as risk scoring, resource forecasting, AI tool adoption, or task assignment—shape real-world Agile outcomes.

This study addresses this gap by employing Random Forest and XGBoost to model the influence of managerial factors on task completion outcomes in an AI-enhanced Agile development environment. Through this approach, the study provides a quantitative understanding of Agile Management effectiveness and contributes a predictive, evidence-based framework for managerial decision-making.

Research Methodology

The study adopts a quantitative and empirical modeling approach to evaluate the effectiveness of Agile management practices using ensemble-based machine learning techniques. The overall methodological flow consists of dataset preparation, preprocessing, model construction, feature extraction, and performance evaluation, all of which are systematically illustrated in the workflow presented in figure 1, which outlines the sequential steps undertaken from raw data to managerial insight generation. This structure ensures transparency, replicability, and analytical rigor throughout the research process.

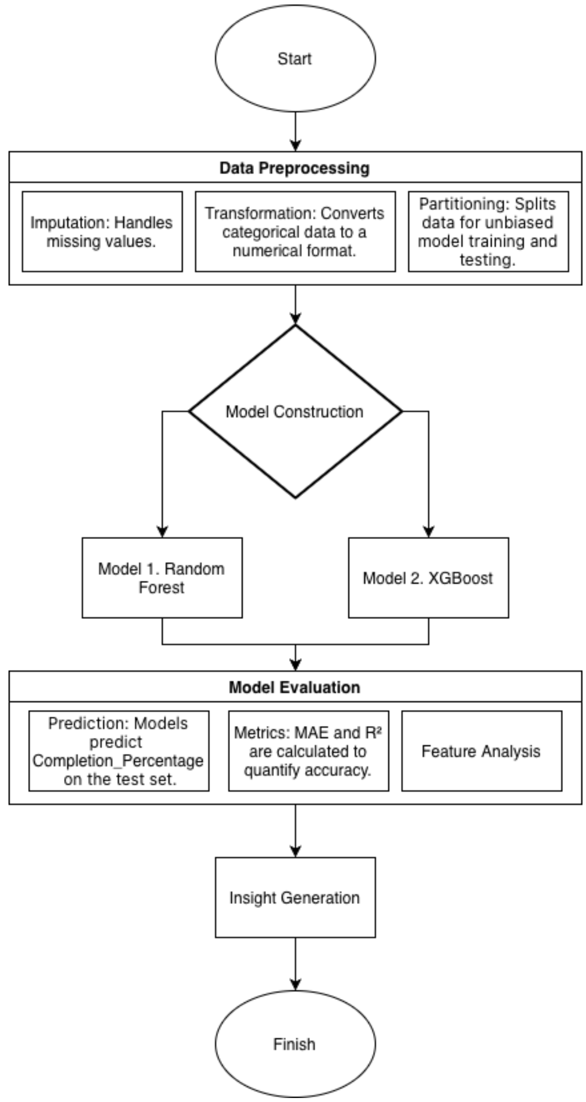

Figure 1 **Research Step**

Research Design

The research applies an empirical design in which managerial and performance variables within Agile development environments are modeled using supervised machine learning. The use of Random Forest and XGBoost supports the detection of nonlinear interactions, robustness against noise, and the generation of interpretable importance rankings. This methodological approach allows for a quantitative assessment of how managerial decisions influence developer task completion outcomes.

Dataset Description

The dataset consists of observations representing managerial inputs, team behaviors, and performance outcomes obtained from AI-assisted Agile development environments. It contains numerical and categorical attributes that capture managerial decisions such as resource distribution, risk evaluation, project placement, and predicted workload requirements. The target variable, completion percentage, reflects the effectiveness of task execution under different managerial conditions.

A summary of the variables included in the dataset is presented in table 1, which categorizes each attribute according to its type, description, and analytic role.

Table 1 Dataset Variable Description

Variable Name	Type	Description	Analytical Role
Experience_Years	Numerical	Number of years of professional development experience	Managerial predictor
Resource_Allocation_Hours	Numerical	Hours allocated to the developer for specific tasks or sprints	Managerial predictor
Risk_Assessment_Score	Numerical	Quantified score representing perceived project or sprint risks	Managerial predictor
Real_Time_Resource_Prediction	Numerical	AI-generated prediction of required resources based on real-time data	Managerial predictor
Project_Overrun_Percentage	Numerical	Percentage of deviation from expected project timelines	Managerial predictor
AI_Optimization_Effectiveness	Numerical	Effectiveness score of AI tools used to support Agile coordination	Managerial predictor
Role	Categorical	Developer's functional position (e.g., frontend, backend, full stack)	Managerial predictor
Project_Assignment	Categorical	Project group or domain to which the developer is assigned	Managerial predictor
Completion_Percentage	Numerical	Percentage of tasks successfully completed (target variable)	Outcome measure

The structure of the dataset reflects the operational realities of Agile environments where managerial decision-making is influenced by workload distribution, team expertise, risk perception, and AI-assisted planning. The variety of managerial variables enables a multi-dimensional analysis of Agile performance.

Data Preprocessing

The preprocessing stage includes the treatment of missing data, the transformation of categorical attributes, and the partitioning of the dataset for model training. Numerical missing values were imputed using the mean, while categorical missing entries were filled using the mode. Label encoding was applied to convert categorical variables into numerical values compatible with tree-based learning algorithms. The dataset was subsequently divided into training and testing subsets using an 80:20 ratio to ensure unbiased model evaluation. The inherent robustness of the chosen algorithms allows the preprocessing pipeline to remain efficient while maintaining data fidelity.

Machine Learning Models

Random Forest is used because of its capacity to model nonlinearities and interactions while maintaining interpretability. It constructs multiple decision trees and aggregates their predictions through averaging. The prediction function is expressed as:

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x) \quad (1)$$

This ensemble formulation allows Random Forest to achieve strong generalization by reducing variance across individual decision trees.

XGBoost serves as the second predictive model due to its scalability and high performance on structured datasets. It constructs trees sequentially by minimizing a regularized objective function:

$$Obj = \sum_{i=1}^{n} l(\widehat{y_i}, y_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (2)$$

The regularization term is defined as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum w_j^2 \quad (3)$$

These components ensure that XGBoost effectively penalizes complexity while optimizing predictive accuracy, making it well-suited for managerial performance modeling.

Feature Importance Analysis

Feature importance analysis is performed to identify which managerial variables contribute most significantly to Agile task completion. Random Forest derives importance from impurity reduction, while XGBoost measures gain, which represents the improvement in model accuracy resulting from splits involving a given feature. Cross-model comparison enhances reliability by highlighting variables consistently ranked as influential across both algorithms.

Evaluation Metrics

Two performance metrics were used to evaluate model accuracy: Mean Absolute Error (MAE) and the coefficient of determination (R^2). The Mean Absolute Error is defined as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \widehat{y_i}| \quad (4)$$

The coefficient of determination is calculated as:

$$R^2 = 1 - \frac{\sum (y_i - \widehat{y_i})^2}{\sum \left(y_i - \underline{y}\right)^2} \quad (5)$$

These metrics quantify prediction accuracy and explainable variance, enabling robust evaluation of model performance.

Research Workflow Summary

The complete research workflow illustrated in figure 1 demonstrates the sequential path taken from raw data acquisition to predictive modeling and managerial insight extraction. The workflow underpins the study's systematic approach and ensures methodological coherence across all analytical stages.

Result

This section presents the analytical results derived from the application of Random Forest and XGBoost to evaluate the effectiveness of Agile management practices in predicting developer performance. Managerial variables such as risk assessment, resource allocation, project assignment, and AI-assisted optimization were modeled to determine their relative influence on task completion outcomes. The results are organized into three primary components: descriptive statistics, feature-importance modeling, and managerial factor interpretation.

Descriptive Statistics

A preliminary descriptive statistical analysis was conducted to provide an overview of the managerial and operational characteristics within the dataset. Table 2 summarizes the distribution of experience levels, resource allocation hours, risk assessment scores, and AI optimization effectiveness. The variability observed in several managerial attributes suggests heterogeneous management styles and workload structures among Agile teams, which is consistent with the adaptive nature of Agile practices.

Table 2 Descriptive Statistics of Key Variables

Variable	Count	Mean	Std. Dev	Min	25%	Median	75%	Max
Experience Years	200	7.51	4.11	1.00	4.00	7.00	11.00	14.00
Resource Allocation Hours	200	49.44	16.68	20.00	35.75	49.50	63.25	79.00
Risk Assessment Score	200	0.497	0.295	0.01	0.24	0.495	0.75	1.00
Completion Percentage	200	74.89	14.18	50.23	62.26	75.02	87.25	99.84
Real-Time Resource Prediction	200	49.65	17.80	20.66	34.21	49.76	64.67	79.88
Project Overrun Percentage	200	12.85	7.53	0.16	5.79	13.46	19.49	24.93
AI Optimization Effectiveness	200	0.500	0.298	0.00	0.22	0.515	0.73	1.00

Dataset Summary Grouped by Managerial Factors

To better understand how managerial design influences performance, the dataset was grouped according to two key management attributes: developer role and project assignment. Table 3 presents the aggregated performance indicators based on these categories. The results demonstrate clear performance differentials across both managerial dimensions. Teams with balanced project assignments and consistent role clarity exhibit higher completion percentages, implying that effective managerial structuring enhances Agile execution outcomes.

Table 3 Dataset Summary by Role and Project Assignment

Role	Project Assignment	Experience Years	Resource Allocation Hours	Risk Assessment Score	Completion Percentage	Project Overrun Percentage	AI Optimization Effectiveness
Backend Developer	Finance IT Project	7.50	48.33	0.534	77.69	12.91	0.551
Backend Developer	Healthcare IT Project	6.38	54.00	0.388	72.87	15.43	0.409
Backend Developer	Retail IT Project	7.78	44.33	0.379	71.84	11.43	0.408
DevOps Engineer	Finance IT Project	7.88	50.94	0.516	78.03	13.69	0.533
DevOps Engineer	Healthcare IT Project	7.53	51.53	0.480	74.55	12.57	0.504
DevOps Engineer	Retail IT Project	8.08	48.25	0.481	74.32	11.82	0.504
Frontend Developer	Finance IT Project	5.44	41.22	0.383	73.64	12.05	0.363
Frontend Developer	Healthcare IT Project	6.61	46.39	0.398	74.39	12.78	0.369
Frontend Developer	Retail IT Project	6.47	46.26	0.398	73.88	12.01	0.593
Full-stack Developer	Finance IT Project	9.06	50.50	0.517	78.56	12.96	0.514
Full-stack Developer	Healthcare IT Project	8.28	51.17	0.510	74.45	13.67	0.662
Full-stack Developer	Retail IT Project	7.75	46.25	0.418	73.47	13.57	0.507

Feature Importance: Random Forest

Random Forest was utilized to identify which managerial and operational factors exert the strongest influence on Completion Percentage. Figure 2 illustrates the relative contribution of each variable. Variables such as Project Overrun Percentage, AI Optimization Effectiveness, Real-Time Resource Prediction, Risk Assessment Score, and Resource Allocation Hours emerge as the most influential predictors. These findings highlight that Agile management effectiveness is strongly tied to dynamic resource planning, risk mitigation practices, and the ability to leverage AI-based tools to support team workflows.

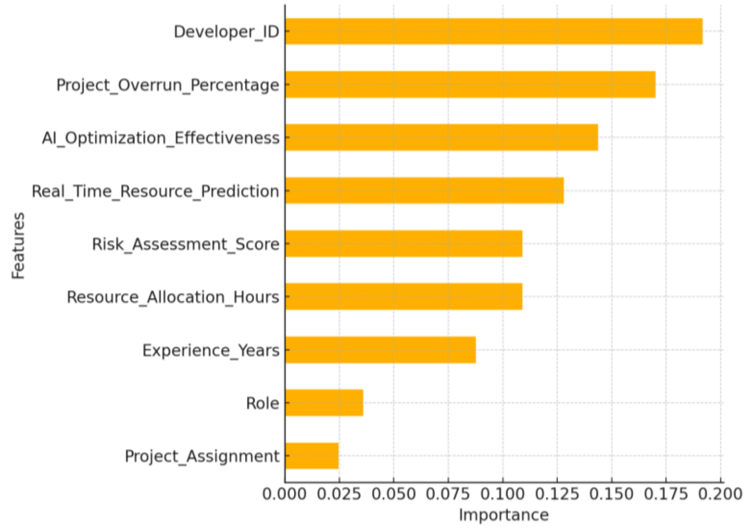

Figure 2 Random Forest Feature Importance

The dominance of overrun and risk-related variables underscores the significance of managerial forecasting accuracy. Effective Agile management reallocates resources early in the sprint, minimizes uncertainty, and reduces task congestion—behaviors that directly increase completion performance.

Feature Importance: XGBoost

The XGBoost model offers complementary managerial insights by identifying Project Assignment, Experience Years, AI Optimization Effectiveness, and Real-Time Resource Prediction as principal determinants of performance. Figure 3 presents the XGBoost feature importance chart.

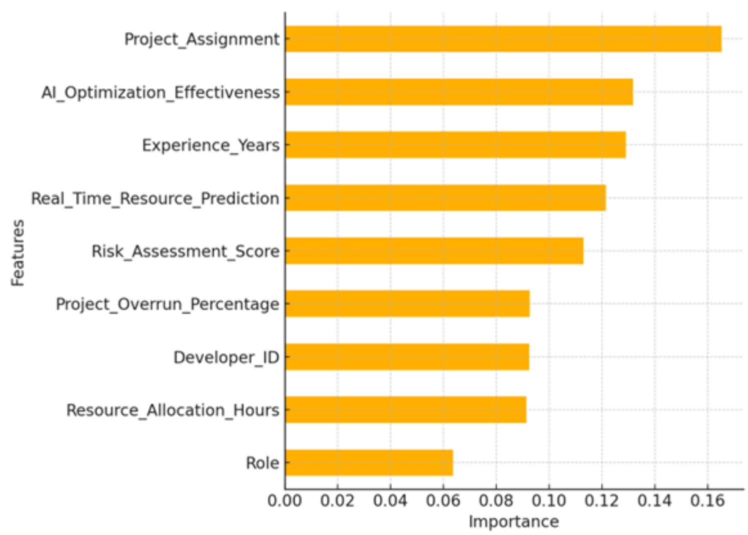

Figure 3 **XGBoost Feature Importance**

XGBoost's prioritization of project assignment underscores the managerial necessity of aligning developer competencies with assigned tasks. Developers with more experience tend to complete a higher proportion of tasks, reinforcing the importance of skill-based workload allocation in Agile environments.

Correlation Analysis of Managerial Variables

A correlation heatmap was generated to examine interdependencies among managerial variables. As shown in figure 4, moderate correlations exist between resource allocation and project overrun, as well as between AI optimization and completion percentages. These patterns further validate the machine learning findings by illustrating how managerial planning, resource management, and tool effectiveness interact to shape Agile productivity outcomes.

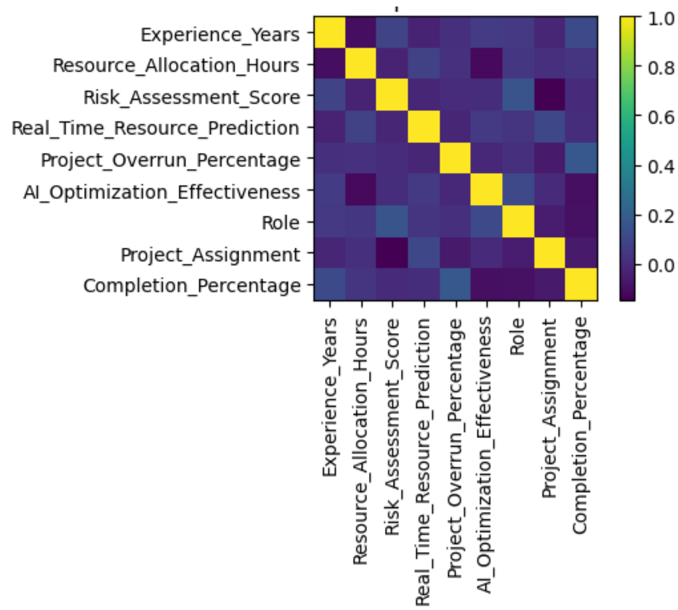

Figure 4 **Correlation Heatmap of Managerial Variables**

Summary of Top Managerial Predictors

To synthesize the results of both models, the feature importances from Random Forest and XGBoost were averaged. Table 4 presents the aggregated ranking. The analysis shows that Project Overrun Percentage, AI Optimization Effectiveness, Resource Prediction Accuracy, Developer Experience, and Task Assignment Quality constitute the most influential managerial factors.

Table 4 **Summary of Top Managerial Predictors**

Managerial Variable	Random Forest Importance	XGBoost Importance	Average Importance	Managerial Interpretation
Project Overrun Percentage	0.168	0.079	0.124	Indicates the effectiveness of sprint planning and managerial control over potential delays.
AI Optimization Effectiveness	0.144	0.113	0.128	Reflects the extent to which AI-assisted tools contribute to improving Agile team performance.

Real-Time Resource Prediction	0.129	0.124	0.127	Measures managerial capability to forecast resource needs adaptively during Agile iterations.
Risk Assessment Score	0.110	0.107	0.108	Represents the quality of risk identification and mitigation processes within Agile practices.
Resource Allocation Hours	0.108	0.083	0.096	Indicates how effectively managers allocate workloads to optimize team productivity.
Experience Years	0.084	0.126	0.105	Highlights the influence of individual expertise and competency on Agile performance outcomes.
Project Assignment	0.025	0.202	0.114	Represents the degree to which managers align developer skills with appropriate project tasks.
Role	0.036	0.070	0.053	Reflects the structural influence of team positions within Agile teams.

This consolidated analysis demonstrates that Agile management effectiveness is strongly driven by the combination of predictive planning, AI-supported optimization, and strategic task allocation at the managerial level.

Interpretation of Model Findings

Taken together, the machine learning outputs reveal a coherent managerial pattern: Agile performance improves significantly when managers are able to foresee potential overruns, allocate resources adaptively, assign tasks based on experience, and integrate AI tools into day-to-day workflows. Both models confirm that Agile management is most effective when decision-making is informed by accurate prediction, real-time monitoring, and alignment between human skillsets and operational demands.

These findings highlight the importance of data-driven managerial intervention in Agile environments, bridging the gap between traditional intuition-based management and modern AI-enhanced decision support.

Discussion

The objective of this study was to evaluate the effectiveness of Agile management practices by examining the influence of various managerial factors on developer performance using Random Forest and XGBoost models. The results provide several important insights into how managerial decisions, resource strategies, and AI-enabled optimization shape performance outcomes within Agile teams.

The machine learning models demonstrated that certain managerial variables consistently exert strong influence on completion performance. Across both algorithms, real-time resource prediction, AI optimization effectiveness, and project overrun percentage emerged as dominant predictors. This finding highlights that Agile performance is highly dependent on the manager's ability to anticipate resource needs, leverage technological tools for workflow optimization, and maintain control over potential schedule deviations. These results reinforce the central principle of Agile: effectiveness arises from adaptability, continuous monitoring, and responsive decision-making.

The strong influence of risk assessment score further underscores the importance of proactive uncertainty management in Agile environments. Teams operating under poorly assessed risks tend to experience reduced task completion efficiency, suggesting that Agile rituals such as sprint planning and retrospectives must be supported by systematic risk evaluation rather than intuition alone. This aligns with prior research emphasizing that risk transparency directly enhances Agile maturity and project success.

The comparative importance of experience years and project assignment—particularly in the XGBoost model—indicates that team composition and task allocation remain essential managerial responsibilities. Assigning tasks that align with developer experience appears to amplify performance, reducing both cognitive load and task switching overhead. This finding implies that Agile teams benefit from deliberate competency-based task assignment rather than equal workload distribution or random allocation. As a result, Agile managers must consider individual expertise as a strategic resource rather than merely an operational variable.

Interestingly, structural factors such as role had relatively weak influence, suggesting that hierarchical distinctions within Agile teams (e.g., backend developer, frontend developer) are less predictive of performance than dynamic factors such as risk, planning accuracy, and resource predictability. This supports the Agile philosophy that cross-functionality and collaboration reduce role-based performance discrepancies.

The correlation analyses and feature importance patterns collectively support the conclusion that Agile Management is most effective when supported by data-driven planning, adaptive resource strategies, and AI-enabled decision-making. The inclusion of AI optimization metrics in the top predictors demonstrates that modern Agile environments increasingly benefit from intelligent tools that improve estimation accuracy, reduce manual overhead, and enhance development flow.

Overall, the discussion underscores that Agile Management effectiveness arises from the interplay between managerial foresight, technology-supported decision-making, and strategic alignment of team capabilities. These findings contribute to Agile literature by providing empirical evidence that performance outcomes can be predicted—and improved—through measurable managerial practices rather than relying solely on qualitative assessment.

Conclusion

This study analyzed the effectiveness of Agile management practices by applying Random Forest and XGBoost models to a dataset of developer performance metrics. The results demonstrate that Agile Management effectiveness can be quantified through machine learning techniques, offering a more objective and systematic evaluation of managerial decisions.

The findings indicate that real-time resource prediction, AI optimization effectiveness, project overrun percentage, and risk assessment score are the most influential managerial factors affecting performance. These insights highlight the critical importance of accurate planning, proactive risk mitigation, and the integration of AI tools into Agile workflows. Additionally, the significance of project assignment and developer experience emphasizes that competency-based task allocation is essential for optimizing performance outcomes.

Through empirical modeling, this research confirms that Agile Management effectiveness extends beyond adherence to Agile rituals and depends on how managers leverage data, technology, and human expertise to guide team operations. Organizations seeking to enhance Agile performance should therefore invest in predictive analytics tools, develop stronger risk management frameworks, and implement structured competency-based assignment strategies.

This study provides a foundation for integrating machine learning as a decision-support mechanism in Agile Management. Future research may expand this work by exploring longitudinal datasets, incorporating behavioral or communication metrics, or examining the role of hybrid AI–human decision frameworks in Agile team environments.

Declarations

Author Contributions

Conceptualization: Q.S.; Methodology: Q.S.; Software: Q.S.; Validation: Q.S.; Formal Analysis: Q.S.; Investigation: Q.S.; Resources: Q.S.; Data Curation: Q.S.; Writing Original Draft Preparation: Q.S.; Writing Review and Editing: Q.S.; Visualization: Q.S.; All authors have read and agreed to the published version of the manuscript.

Data Availability Statement

The data presented in this study are available on request from the corresponding author.

Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. K. Gupta and V. S. Suma, “Agile software development: A survey of trends and techniques,” *International Journal of Computer Applications*, vol. 975, no. 8887, pp. 1–7, 2013, doi: 10.5120/15459-4277.

[2] J. Cho, “Issues and challenges of Agile software development with Scrum,” *Issues*

in Information Systems, vol. 9, no. 2, pp. 188–195, 2008, doi: 10.48009/2_iis_2008_188-195.

[3] M. M. Shastri, D. H. Hsu, and Y. Li, “Impact of Agile methodologies on software development,” *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 5, pp. 111–116, 2015, doi: 10.14569/IJACSA.2015.060515.

[4] Sidky and J. Arthur, “Determining the readiness of organizations for Agile software development,” *Information and Software Technology*, vol. 49, no. 8, pp. 719–734, 2007, doi: 10.1016/j.infsof.2007.02.004.

[5] P. Kruchten, R. Nord, and I. Ozkaya, “Technical debt: From metaphor to theory and practice,” *IEEE Software*, vol. 29, no. 6, pp. 18–21, 2012, doi: 10.1109/MS.2012.167.

[6] S. Kargar and A. R. Shahriari, “A predictive analytics approach to project performance management in Agile software development,” *IEEE Access*, vol. 8, pp. 227450–227462, 2020, doi: 10.1109/ACCESS.2020.3044983.

[7] Mabu, C. Djemame, and J. Padget, “Machine learning-based workload prediction in cloud and software engineering environments,” *Future Generation Computer Systems*, vol. 87, pp. 41–52, 2018, doi: 10.1016/j.future.2018.04.064.

[8] Hajjdiab and A. Taleb, “Adopting Agile software development: The relevance of organizational culture,” *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 2, pp. 105–111, 2011, doi: 10.14569/IJACSA.2011.020216.

[9] F. Paasivaara, C. Lassenius, V. Heikkilä, and A. Mäkinen, “Challenges and success factors of large-scale Agile transformations: A multiple-case study,” *Empirical Software Engineering*, vol. 23, no. 1, pp. 2550–2593, 2018, doi: 10.1007/s10664-017-9526-3.

[10] K. Beck, M. Beedle, A. van Bennekum, et al., Manifesto for Agile Software Development, Agile Alliance, 2001. https://agilemanifesto.org/

[11] K. Schwaber and J. Sutherland, The Scrum Guide, Scrum Alliance and Scrum.org, 2020. https://scrumguides.org/

[12] T. Dingsøyr, S. Nerur, V. Balijepally, and N. B. Moe, “A decade of agile methodologies: Towards explaining agile software development,” *Journal of Systems and Software*, vol. 85, no. 6, pp. 1213–1221, 2012, doi: 10.1016/j.jss.2012.02.033.

[13] K. Conboy, “Agility from first principles: Reconstructing the concept of agility in information systems development,” *Information Systems Research*, vol. 20, no. 3, pp. 329–354, 2009, doi: 10.1287/isre.1090.0236.

[14] N. B. Moe, D. Šmite, P. J. Ågerfalk, and P. Rodríguez, “Agile teams and autonomy in large-scale agile,” *IEEE Software*, vol. 37, no. 2, pp. 58–63, 2020, doi: 10.1109/MS.2019.2934055.

[15] C. J. Stettina and J. Hörz, “Agile portfolio management: An empirical perspective on the practice in use,” *International Journal of Project Management*, vol. 33, no. 1, pp. 140–152, 2015, doi: 10.1016/j.ijproman.2014.03.008.

[16] V. Stray, N. B. Moe, and A. Aurum, “Coordination mechanisms for cross-team interactions in software development scaling,” *Journal of Systems and Software*,

vol. 187, pp. 111214-1–111214-12, 2022, doi: 10.1016/j.jss.2021.111214.

[17] P. Serrador and J. K. Pinto, “Does Agile work? A quantitative analysis of Agile project success,” *International Journal of Project Management*, vol. 33, no. 5, pp. 1040–1051, 2015, doi: 10.1016/j.ijproman.2015.01.006.

[18] L. Leite, L. Rocha, F. Kon, S. Meira, and D. Milojicic, “A survey of DevOps concepts and challenges,” *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–35, 2019, doi: 10.1145/3359981.

[19] J. Wan, H. Cai, K. Zhou, et al., “Software-defined industrial Internet of Things in the context of Industry 4.0,” *IEEE Sensors Journal*, vol. 16, no. 20, pp. 7373–7380, 2016, doi: 10.1109/JSEN.2016.2565621.

[20] S. Amershi, C. Bird, J. Salgado, et al., “Software engineering for machine learning: A case study,” in *Proc. IEEE/ACM Int. Conf. Software Engineering (ICSE)*, 2019, pp. 291–300, doi: 10.1109/ICSE-SEIP.2019.00047.

[21] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[22] M. Kuhn and K. Johnson, “Applied Predictive Modeling”, New York, NY, USA: Springer, 2013, doi: 10.1007/978-1-4614-6849-3.

[23] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[24] D. Radjenović, M. Heričko, R. Torkar, and A. Živkovič, “Software fault prediction metrics: A systematic literature review,” *Information and Software Technology*, vol. 55, no. 8, pp. 1397–1418, 2013, doi: 10.1016/j.infsof.2013.02.009.