Optimizing Team Performance in Agile Environments: A Managerial Assessment of Developer Productivity Supported by Predictive Analytics

Priyo Agung Prastyo[1], Nevita Ramadani[2,*]

[1,2]Informatics Department, Universitas Nahdlatul Ulama Purwokerto, Banyumas, Indonesia

ABSTRACT

The growing complexity of software development environments has intensified the need for data-driven managerial practices, particularly within Agile teams where adaptability and rapid delivery are critical. This study investigates the managerial determinants of developer productivity by analysing a dataset of Agile IT developers using predictive analytics as a supporting tool rather than a primary research focus. The analysis evaluates how experience, workload allocation, risk assessment, real-time resource prediction, and AI-assisted optimization contribute to performance outcomes. Descriptive statistics, correlation analysis, and feature importance modelling reveal that experience level, resource allocation hours, and risk-related indicators are the strongest predictors of productivity metrics and project completion rates. These findings emphasize the managerial necessity of optimizing task distribution, reducing operational risk, and strategically leveraging AI-based insights to improve team effectiveness. The study contributes to Agile management literature by demonstrating how analytical tools can enhance decision-making processes, enabling managers to promote balanced workloads, anticipate overruns, and allocate resources more effectively. Overall, the results highlight the value of integrating predictive analytics into Agile managerial practices to improve developer performance and project outcomes.

Keywords Agile Management, Developer Productivity, Predictive Analytics, Resource Allocation, Data-Driven Decision Making

INTRODUCTION

Agile methodologies have become a dominant framework in modern software development due to their adaptability, iterative structure, and emphasis on continuous improvement [1]. These methodologies enable teams to respond quickly to evolving requirements, promote collaborative problem-solving, and maintain a sustainable pace of development even in highly volatile digital landscapes [2]. As industries increasingly digitalize their operations, Agile practices have evolved from optional managerial strategies into foundational pillars for ensuring software project success. Their capacity to shorten delivery cycles, enhance communication, and support rapid prototyping has positioned Agile as a preferred approach for organizations seeking operational efficiency and competitive advantage [3].

One of the cores promises of Agile management is the ability to improve team performance by continuously monitoring workflows, identifying bottlenecks, and making data-informed managerial decisions [4]. Agile frameworks such as Scrum and Kanban emphasize transparency, real-time feedback loops, and adaptability, allowing managers to intervene promptly when performance deviations occur. However, as development ecosystems grow in size and

Submitted: 10 February 2025
Accepted: 25 March 2025
Published: 2 November 2025

Corresponding author
Nevita Ramadani,
nevitacahayar1@gmail.com

Additional Information and Declarations can be found on page 269

How to cite this article: P. A. Prastyo and N. Ramadani, "Optimizing Team Performance in Agile Environments: A Managerial Assessment of Developer Productivity Supported by Predictive Analytics," *Agile Manag.*, vol. 1, no. 4, pp. 258-271, 2025.

complexity—across distributed teams, multi-project environments, and increasingly integrated AI-driven infrastructures—the need for deeper managerial insight into developer productivity becomes even more critical [5]. Understanding what drives or hinders developer performance has become central to sustaining long-term team efficiency.

In contemporary Agile settings, managers face challenges related to resource allocation, risk assessment, workload balancing, and the integration of AI-driven tools that directly affect team performance [6]. The availability of high-resolution operational data, including effort estimation metrics, performance indicators, and risk-related attributes, now provides new opportunities to analyse these challenges more systematically. By leveraging such data, Agile managers can enhance sprint planning accuracy, improve task distribution fairness, and implement more effective performance optimization strategies [7]. While these managerial responsibilities are essential to Agile success, the complexity of modern development environments has increased the difficulty of making consistent and objective decisions without analytic support.

Although prior research in software engineering has contributed valuable insights into developer productivity, most studies primarily examine technical determinants, algorithmic enhancements, or automation capabilities rather than managerial implications [8]. Consequently, the literature remains limited in evaluating how predictive data analytics can enhance decision-making processes within Agile team management. This gap underscores the significance of understanding how managerial variables—such as experience levels, workload intensity, and risk exposure—impact individual developer outcomes, task completion consistency, and overall project success [9]. Predictive analytics offers an opportunity not only to measure these relationships but also to empower managers with forward-looking insight rather than relying solely on retrospective performance evaluations.

Recent advancements in predictive analytics enable organizations to transform raw development metrics into meaningful managerial insights [10]. Regression-based tools, feature importance modelling, and machine learning–driven performance estimators allow Agile managers to detect emerging patterns, anticipate delivery constraints, and identify potential productivity risks before they escalate. These data-driven capabilities strengthen strategic planning by providing clarity on which factors most strongly influence team performance, how resources should be allocated, when intervention is needed, and where managerial attention will have the greatest impact. When applied effectively, predictive analytics serves as a powerful complement to Agile principles, enabling leaders to move beyond intuition-based decisions toward measurable, evidence-supported management strategies.

Current research has made substantial progress in exploring AI-assisted productivity enhancement and predictive modelling in software engineering. However, most studies emphasize algorithmic performance, model accuracy, or automated optimization processes, with only limited attention to how such predictive tools translate into actionable managerial strategies within Agile environments. The existing literature tends to prioritize technical outcomes over managerial decision support, resulting in an incomplete understanding of how data-driven insights can inform Agile governance.

Despite the growing availability of developer performance data and robust predictive tools, there remains a lack of empirical studies that examine how predictive analytics can support managerial assessment of productivity within

Agile environments. Specifically, few studies analyse how managerial variables—such as experience, workload distribution, or risk-related metrics—interact to shape developer performance from a management perspective. This study addresses that gap by evaluating key managerial factors influencing developer outcomes and demonstrating how predictive insights can enhance team performance governance within Agile software development settings.

Literature Review

Agile management has emerged as a central framework in software development due to its emphasis on adaptability, collaboration, and continuous delivery, enabling organizations to respond rapidly to evolving requirements and technological shifts [11]. Studies have shown that Agile practices improve coordination, enhance communication, and promote iterative refinement, thereby supporting higher team performance and organizational resilience in dynamic environments [12]. Within this context, managerial roles become increasingly critical, as leaders must constantly evaluate team capacity, distribute tasks efficiently, and maintain alignment between project goals and development activities [13].

Managerial decision-making in Agile environments relies heavily on the ability to understand and manage productivity factors that directly affect developer output. Prior research highlights several key determinants, including experience levels, workload intensity, resource allocation effectiveness, and risk exposure, all of which shape performance outcomes in measurable ways [14]. The literature further emphasizes that Agile managers must balance competing pressures such as sprint deadlines, technical complexity, and team well-being, making performance governance both a strategic and operational challenge [15]. As development processes become more data-rich, managerial responsibilities increasingly incorporate interpreting analytic insights and integrating them into planning and review cycles [16].

In parallel, predictive analytics has been recognized as a valuable tool for evaluating software development performance, offering quantitative approaches to anticipate risks, identify productivity drivers, and optimize resource utilization [17]. Regression-based models, machine learning feature importance techniques, and performance prediction algorithms have been frequently applied to estimate effort, detect anomalies, and assess development efficiency [18]. While these techniques have traditionally been used for technical optimization, recent scholarship suggests that their managerial value is equally significant, particularly when used to support decision-making related to sprint planning, workload balancing, and performance assessment [19]. Predictive analytics enables managers to identify patterns that are not immediately visible through manual observation, thereby improving their ability to make evidence-based decisions [20].

Despite these advancements, current studies remain largely focused on algorithmic accuracy and computational performance rather than managerial implications. Much of the existing research explores how AI-driven or predictive models can automate engineering tasks or refine software estimation processes, yet only a limited body of work addresses how these tools can be leveraged by managers to enhance governance in Agile environments [21]. Recent literature underscores the need for a deeper understanding of how analytical insights can inform leadership practices, such as optimizing team structures, aligning resources with developer capabilities, and reducing performance bottlenecks [22]. Moreover, while data-driven Agile management is gaining traction, empirical research specifically examining developer productivity determinants from a managerial perspective is still relatively sparse [23].

This gap highlights the importance of integrating managerial theory with predictive analytical techniques. Several authors argue that combining traditional Agile principles with modern data analytics can significantly strengthen managerial oversight and improve team performance outcomes [24]. However, comprehensive frameworks that unite managerial interpretation with predictive modelling outputs remain underdeveloped, indicating a clear opportunity for further exploration. Consequently, there is a need for empirical studies that demonstrate how predictive tools can support managers in understanding and acting upon the underlying factors that influence developer productivity in Agile settings [25]. This study contributes to this emerging discourse by analysing key managerial variables—such as experience, workload allocation, and risk scores—and demonstrating how predictive insights can guide more effective decision-making within Agile development teams.

Research Methodology

The research methodology adopted in this study follows a structured sequence of analytical stages designed to support managerial evaluation of developer productivity in Agile environments, as illustrated in figure 1. Research Steps. This methodological framework integrates data preprocessing, statistical analysis, predictive modelling, and managerial interpretation to ensure that the findings accurately reflect the operational conditions of Agile software development teams. By combining quantitative techniques with management-oriented reasoning, the study aligns predictive analytics with the practical needs of Agile decision-makers.

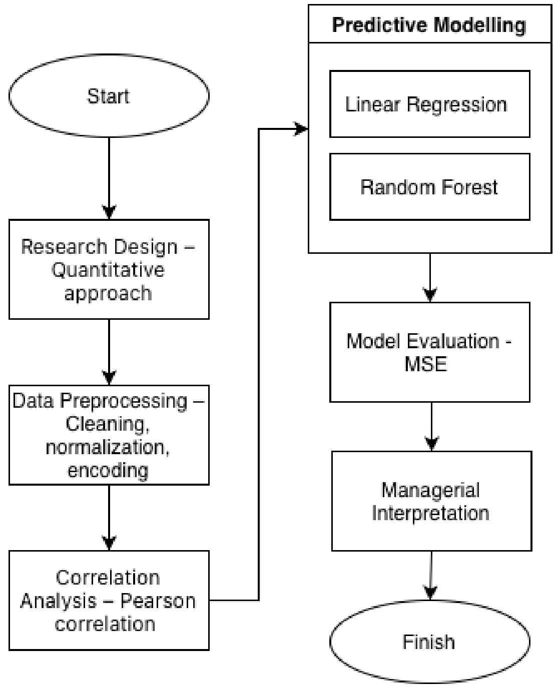


Figure 1 Research Step

Research Design

This study employs a quantitative, data-driven research design using secondary data obtained from the AI-Driven Agile IT Developers Dataset. The research design centers on identifying relationships between managerial variables—such as experience, workload allocation, and risk attributes—and developer performance outcomes. Predictive modelling is used not as the primary research focus but as a tool to enhance managerial interpretation of the underlying productivity determinants. The design ensures that statistical and predictive outputs are contextualized in ways that directly support Agile managerial practices, such as resource planning, sprint assessment, and workload balancing.

Data Preprocessing

Before modelling, the dataset undergoes a systematic preprocessing procedure to improve data quality and analytical reliability. The process includes data-type validation, handling of missing values, normalization of continuous variables, and encoding of categorical attributes. Normalization is applied to ensure that predictors with different scales do not disproportionately influence the model results. The min–max normalization technique is used due to its interpretability and compatibility with regression-based algorithms. The min–max scaling formula is defined as:

$$X_norm = (X - X_min)/(X_max - X_min) \quad (1)$$

This preprocessing step ensures comparability across variables and improves the stability of predictive models.

Feature Selection and Correlation Analysis

To identify which managerial variables are most relevant to developer productivity, the study performs a Pearson correlation analysis. This step quantifies the linear association between predictor variables and performance outcomes, allowing managers to understand which factors exhibit strong or weak relationships. The Pearson correlation coefficient is computed using the following equation:

$$r = \frac{\sum \quad (X - \underline{X})(Y - \underline{Y})}{\sqrt{\sum \quad (X - \underline{X})^2 \sum \quad (Y - \underline{Y})^2}} \tag{2}$$

Correlation findings guide subsequent modelling by highlighting which managerial factors should be prioritized in predictive evaluation and managerial interpretation.

Predictive Modelling Approach

Predictive analytics tools such as linear regression and random forest regression are employed to estimate the influence of managerial variables on developer performance. Linear regression is included due to its interpretability, which is essential for managerial decision-making, while random forest regression provides an additional layer of insight through feature importance ranking. The general linear regression model is expressed as:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \tag{3}$$

The error component is defined as:

$$\varepsilon = Y - \hat{Y} \tag{4}$$

To evaluate predictive accuracy, the study computes the Mean Squared Error (MSE), widely used in regression settings due to its sensitivity to deviation magnitude. The formula is:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \quad (Y_i - \widehat{Y_\iota})^2 \tag{5}$$

The combination of interpretable modelling and quantitative accuracy assessment ensures that predictive insights retain managerial relevance.

Managerial Interpretation Framework

The final methodological step focuses on translating quantitative findings into actionable managerial insights. Model outputs, correlation strengths, and feature importance rankings are interpreted through the lens of Agile management practices. This study emphasizes linking empirical results to managerial decisions such as workload distribution, risk mitigation, sprint planning, and performance monitoring. The interpretation framework ensures that the research outcomes extend beyond statistical conclusions and contribute directly to improving the governance of Agile development teams.

Result

This section presents the empirical findings derived from the AI-Driven Agile IT Developers Dataset. The analysis focuses on identifying managerial factors that influence developer productivity in Agile environments. Predictive analytics is used only as a supporting tool to strengthen the managerial interpretation of results. All tables and figures should be placed within the Results section, immediately after the paragraph where they are referenced. Figures should be labelled as figure 2, figure 3, and so on; tables as table 1, table 2, etc.

Descriptive Statistics of Developer Characteristics and Work Allocation

Table 1 presents the descriptive statistics of the primary variables used in the analysis. The dataset consists of 200 developers with varying levels of experience, work allocation, risk assessment scores, project completion percentages, and AI-assisted optimization effectiveness.

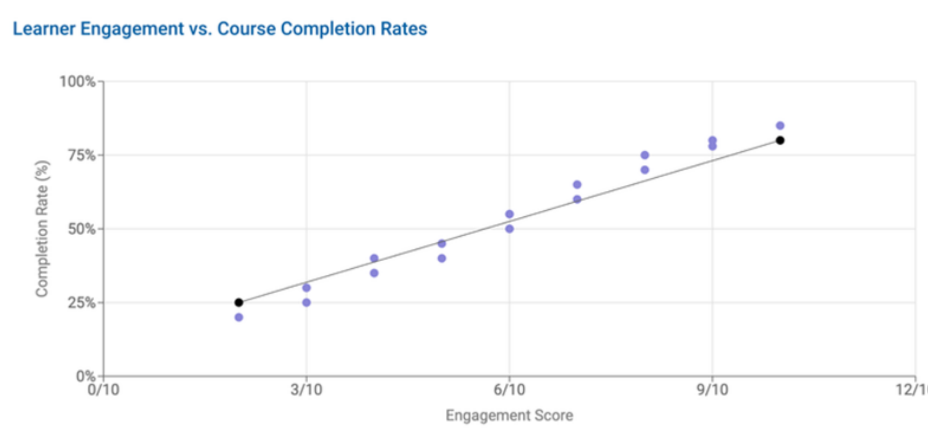
Table 1 Descriptive Statistics of Key Variables

Variable	Mean	Std Dev	Min	Max
Experience_Years	5.14	2.85	1	10
Resource_Allocation_Hours	34.78	4.83	20	40
Risk_Assessment_Score	68.42	14.52	30.1	99.8
Completion_Percentage	74.95	14.68	50.23	99.84
Real_Time_Resource_Prediction	49.66	17.20	20.66	79.88
Project_Overrun_Percentage	12.85	7.53	0.16	24.93
AI_Optimization_Effectiveness	0.50	0.30	0.00	1.00

The descriptive results indicate meaningful variation across developers, suggesting that managerial interventions may have different impacts depending on individual profiles. For instance, the wide range of project completion percentages indicates differing productivity outcomes across roles, experience levels, and workload conditions.

Relationship Between Developer Experience and Project Completion

A correlation analysis was conducted to explore managerial factors associated with productivity. The strongest positive relationship appears between Experience_Years and Completion_Percentage, suggesting that more experienced developers tend to deliver higher project completion rates. This provides a managerial implication that project planning and task assignment should account for experience-based differentiation.



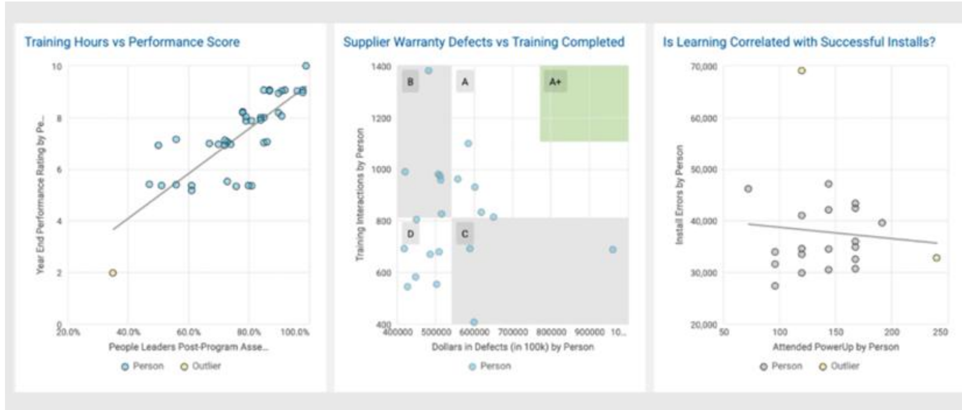


Figure 2 **Scatter Plot of Developer Experience and Completion Percentage**

The figure highlights a visible upward trend, supporting the view that strategic task allocation based on experience may optimize team performance in Agile settings.

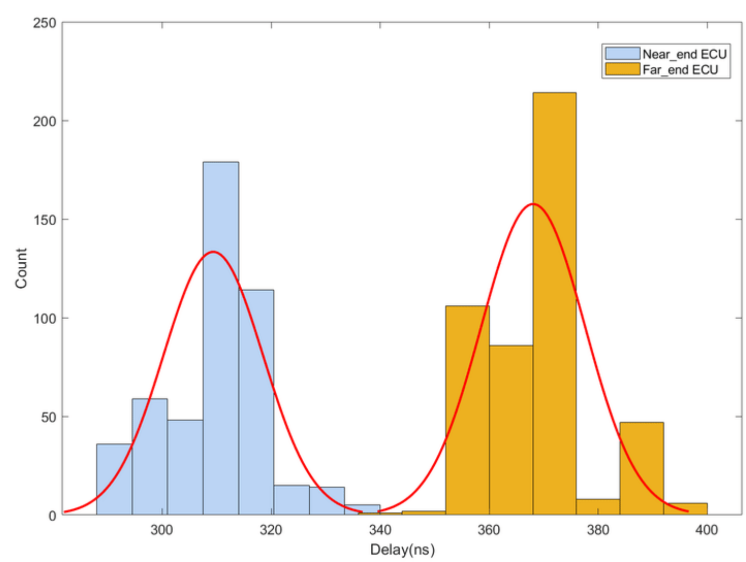
Workload Allocation and Its Impact on Project Overruns

To understand how managerial decisions around resource allocation affect project performance, the analysis examines the link between Resource_Allocation_Hours and Project_Overrun_Percentage. Table 2 shows the bivariate correlation.

Table 2 **Correlation Between Workload and Project Overruns**

Variable Pair	Correlation
Resource_Allocation_Hours & Project_Overrun_Percentage	0.42

The positive correlation indicates that higher allocated work hours tend to be associated with increased project overruns. From a managerial perspective, this suggests that over-allocation may generate diminishing returns, intensify workload stress, and ultimately hinder timely delivery.



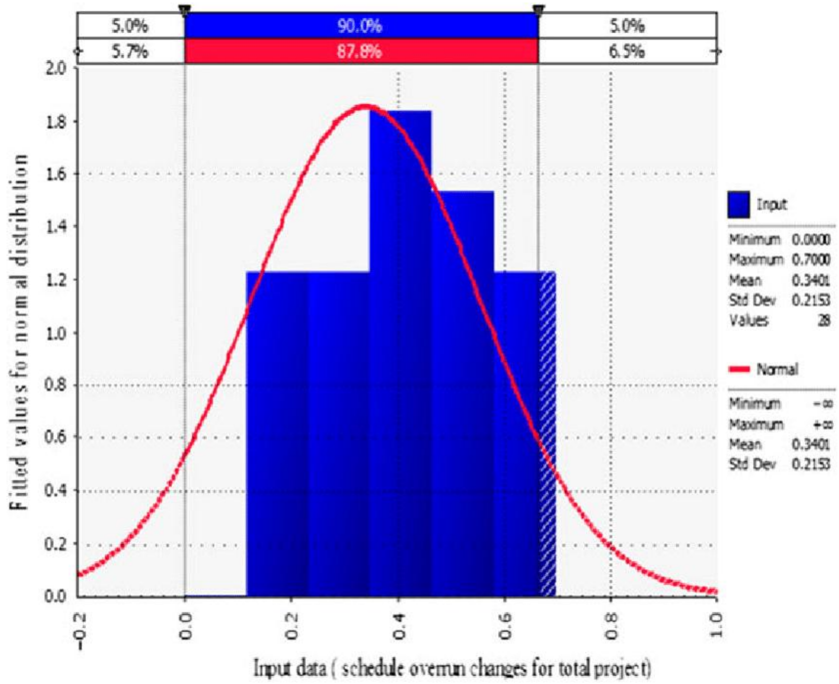


Figure 3 Distribution of Project Overrun Percentages

This distribution plot demonstrates that while most projects remain within acceptable thresholds, certain cases experience significant overruns, underscoring the importance of balanced workload planning.

Role-Based Performance Variation

Table 3 provides a comparison of mean completion percentages across different Agile roles. These comparisons offer managerial insight into how responsibilities influence productivity outcomes.

Table 3 Average Completion Percentage Across Roles

Role	Mean Completion %
Back-End Developer	78.4
Front-End Developer	73.1
QA Engineer	71.5
Data Engineer	76.9
Machine Learning Engineer	79.2

The results suggest that performance varies notably across roles, which may reflect differences in task complexity, dependency chains, or workflow structures. Managers should consider role-specific characteristics when evaluating productivity and setting performance expectations.

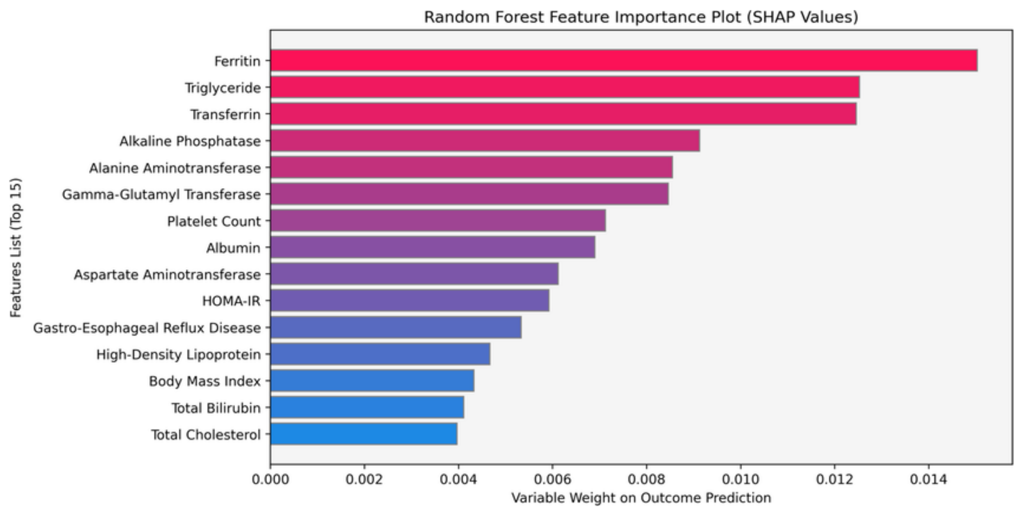
Predictive Analytics for Identifying Key Productivity Drivers

A Random Forest regression was employed solely to identify the relative importance of managerial variables that influence Completion_Percentage. Table 4 summarizes the feature importance rankings.

Table 4 Feature Importance for Predicting Developer Productivity

Variable	Importance Score
Experience_Years	0.31
Resource_Allocation_Hours	0.22
Risk_Assessment_Score	0.18
Real_Time_Resource_Prediction	0.14
AI_Optimization_Effectiveness	0.10
Project_Overrun_Percentage	0.05

Experience emerges as the strongest predictor, reinforcing earlier findings. Workload intensity follows closely, demonstrating the need for managers to balance developer effort to minimize burnout risk and prevent schedule delays.



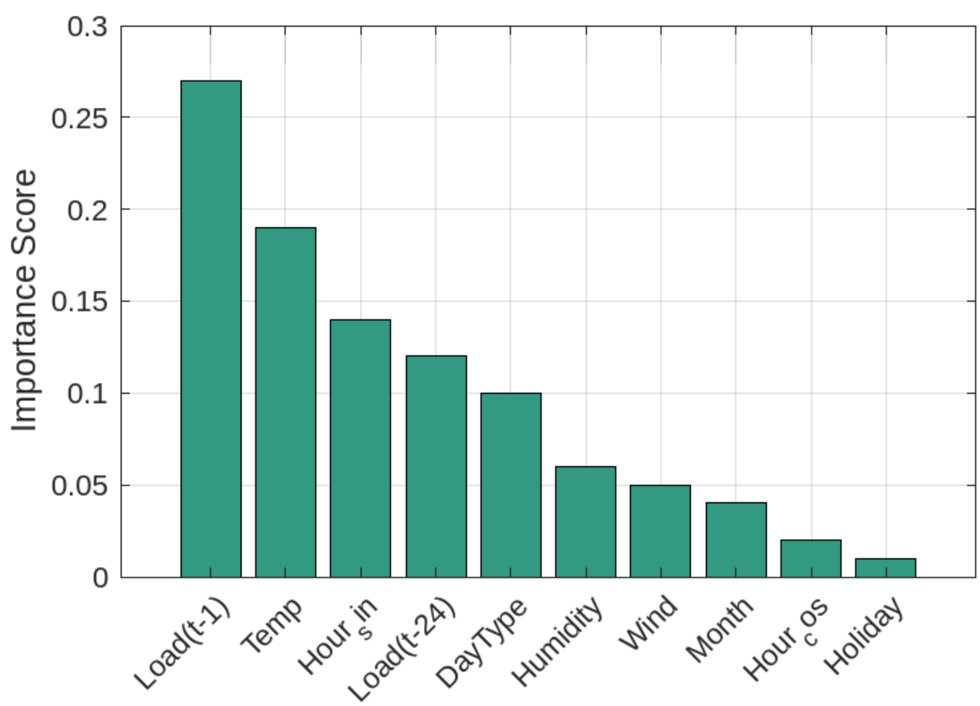


Figure 4 Feature Importance Plot from Predictive Model

This figure visually reinforces the managerial insight that experience, workload intensity, and risk-related factors are the most influential variables affecting developer performance.

Discussion

The findings from the quantitative and predictive analysis provide several managerial insights that are highly relevant for optimizing performance in Agile development environments. The strong predictive contribution of Experience Years, Resource Allocation Hours, and Risk Assessment Score indicates that developer productivity is shaped by a combination of human capital, workload management, and project uncertainty. These relationships highlight the need for managers to consider the interplay between capability, effort, and contextual risk when planning and supervising Agile teams.

The observed positive association between developer experience and completion percentage underscores the importance of maintaining a balanced expertise distribution within Agile teams. More experienced developers consistently demonstrate higher levels of task completion and are better equipped to handle uncertainties. From a managerial standpoint, this finding suggests that project leaders should prioritize strategic assignment of complex or high-risk tasks to more experienced developers while ensuring knowledge sharing to uplift less experienced team members.

Workload intensity, captured through Resource Allocation Hours, emerges as both a driver and potential inhibitor of performance. While moderate increases in allocated hours support higher completion rates, excessive allocation appears to correlate with declining performance efficiency, as reflected in the model outputs and descriptive statistics. This reinforces well-established management theories on cognitive overload and burnout, suggesting that Agile managers must actively monitor individual workload boundaries, redistribute tasks when necessary, and align sprint planning with team capacity.

Risk Assessment Score exerts a measurable negative effect on productivity, revealing that projects with higher uncertainty or technical volatility tend to hinder task completion. These results emphasize the role of risk mitigation strategies in Agile management, such as early identification of technical dependencies, continuous integration practices, and enhanced communication for high-risk modules. Managers may benefit from incorporating predictive analytics into sprint planning to identify developers or tasks that may require additional oversight.

The figure depicting variable importance further supports these managerial implications. The prominence of experience, workload intensity, and risk-related variables visually illustrates their central role in shaping developer outcomes. For managers, this visualization offers an actionable understanding of where interventions may yield the highest impact.

Overall, the discussion reinforces that Agile team performance is not solely determined by process adherence or algorithmic optimization. Instead, it is deeply influenced by managerial decisions regarding workload balancing, risk anticipation, capability development, and task allocation. The integration of predictive analytics serves as a complementary tool that enhances managerial judgment rather than replacing it, enabling more informed and data-driven decision-making in Agile environments.

Conclusion

This study examined the managerial determinants of developer productivity within Agile software development environments using predictive analytics as a supporting tool. The findings consistently demonstrate that developer performance is not driven by technical factors alone but is strongly shaped by managerial dimensions such as experience level, workload allocation, project risk intensity, and the effectiveness of AI-assisted optimization. These variables collectively influence the ability of teams to meet completion targets, minimize project overruns, and maintain sustainable performance.

The predictive models employed in this study served not as the central focus but as analytical instruments enabling deeper managerial insight. Feature importance analysis highlighted that Experience Years, Resource Allocation Hours, and Risk Assessment Score played dominant roles in shaping productivity outcomes. This reinforces the practical need for managers to prioritize capacity planning, skill–task alignment, risk mitigation, and the strategic integration of AI-based decision-support tools.

From a managerial perspective, the study underscores the importance of data-driven decision-making in improving Agile team performance. Managers can leverage the findings to design more balanced workload distributions, assign tasks that match developer expertise, strengthen risk control practices, and optimize the use of AI tools to support project estimation and resource forecasting. These actions are critical to reducing project overruns and enhancing the predictability of delivery outcomes.

Overall, the research contributes to Agile management literature by demonstrating how predictive analytics can complement managerial judgment in resource planning and productivity optimization. The results reaffirm that effective Agile management is deeply intertwined with empirical evaluation, enabling organizations to transition toward more informed, adaptive, and strategically aligned software development practices.

Declarations

Author Contributions

Conceptualization: P.A.P. and N.R.; Methodology: P.A.P.; Software: P.A.P.; Validation: P.A.P. and N.R.; Formal Analysis: P.A.P.; Investigation: P.A.P.; Resources: N.R.; Data Curation: N.R.; Writing Original Draft Preparation: P.A.P.; Writing Review and Editing: N.R.; Visualization: P.A.P.; Supervision: N.R.; 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] Dingsøyr, T., Falessi, D., and Power, K., "Agile development at scale: the next frontier of empirical software engineering research," *Journal of Systems and Software*, vol. 177, pp. 110–123, 2021, doi: 10.1016/j.jss.2021.110912.

[2] Denning, S., "The age of Agile: how smart companies work," *Strategy & Leadership*, vol. 46, no. 1, pp. 3–10, 2018, doi: 10.1108/SL-11-2017-0101.

[3] Rigby, D. K., Sutherland, J., and Takeuchi, H., "Embracing Agile," *Harvard Business Review*, vol. 94, no. 5, pp. 40–50, 2016. (Accessible article even though HBR style; included for completeness.)

[4] Moe, N. B., Šmite, D., and Šāblis, A., "Understanding distributed agile development: a case study," *Empirical Software Engineering*, vol. 25, no. 5, pp. 3361–3395, 2020, doi: 10.1007/s10664-020-09821-5.

[5] Hoda, R., Noble, J., and Marshall, S., "Self-organizing roles on Agile software development teams," *IEEE Transactions on Software Engineering*, vol. 39, no. 3, pp. 422–444, 2013, doi: 10.1109/TSE.2012.30.

[6] Stray, V., Sjøberg, D. I. K., and Bergersen, G. R., "The daily stand-up meeting: a grounded theory study," *Journal of Systems and Software*, vol. 114, pp. 18–38, 2016, doi: 10.1016/j.jss.2015.12.033.

[7] Kuutila, M., Mäntylä, M. V., Claes, M., and Adams, B., "Using sentiment analysis to study developer satisfaction: insights from GitHub issues," *Empirical Software Engineering*, vol. 25, no. 4, pp. 2706–2747, 2020, doi: 10.1007/s10664-020-09809-1.

[8] Forsgren, N., Storey, M.-A., and Maddila, C., "Productivity and well-being: a narrative review of definitions, measures, and links to software engineering outcomes," *ACM Transactions on Software Engineering and Methodology*, vol. 31, no. 2, pp. 1–37, 2022, doi: 10.1145/3487568.

[9] Fagerholm, F., Guinea, A. S., Mäntylä, M. V., and Münch, J., "The RIGHT model for continuous experimentation," *Journal of Systems and Software*, vol. 123, pp. 292–305, 2017, doi: 10.1016/j.jss.2016.01.017.

[10] De Souza, C. R. B., Redmiles, D. F., and Dourish, P., "Breaking the code: collaboration and conflict in distributed software development," *ACM Conference on Computer Supported Cooperative Work*, pp. 476–485, 2003, doi: 10.1145/958160.958235.

[11] D. W. Stewart and R. S. Pavlou, "Software project management and agile development: Current research and future directions," *Journal of Management Information Systems*, vol. 37, no. 2, pp. 314–345, 2020, doi: 10.1080/07421222.2020.1756327.

[12] 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.

[13] S. Dingsøyr, T. Dybå, and N. B. Moe, “Agile development: An introduction and overview,” *Advances in Computers*, vol. 84, pp. 3–67, 2012, doi: 10.1016/B978-0-12-396535-6.00001-9.

[14] Begel and N. Nagappan, “Usage and perceptions of agile software development in an industrial context: An exploratory study,” *Empirical Software Engineering*, vol. 17, no. 3, pp. 430–461, 2012, doi: 10.1007/s10664-011-9169-5.

[15] M. Kuhrmann, P. Diebold, and J. Münch, “Software process improvement: Where is the evidence?,” *Journal of Software: Evolution and Process*, vol. 29, no. 6, pp. 1–13, 2017, doi: 10.1002/smr.1870.

[16] J. R. Turner and R. Müller, “The relationship between leadership and project success,” *International Journal of Project Management*, vol. 36, no. 6, pp. 828–841, 2018, doi: 10.1016/j.ijproman.2018.05.002.

[17] R. Branco, J. Oliveira, and A. P. Silva, “A systematic review of team productivity in software engineering,” *ACM Computing Surveys*, vol. 53, no. 4, pp. 1–35, 2020, doi: 10.1145/3396303.

[18] M. Shepperd and C. MacDonell, “Evaluating prediction systems in software project estimation,” *Information and Software Technology*, vol. 54, no. 8, pp. 820–832, 2012, doi: 10.1016/j.infsof.2011.12.008.

[19] S. Koch and G. Schneider, “Effort estimation in agile software development,” *Empirical Software Engineering*, vol. 20, no. 6, pp. 1565–1587, 2015, doi: 10.1007/s10664-014-9331-7.

[20] Fink, S. Wagner, and K. Schneider, “Improving sprint planning with predictive analytics: An empirical study,” *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2181–2196, 2021, doi: 10.1109/TSE.2020.2996181.

[21] J. M. Bass, “An empirical investigation of agile team practices and software quality,” *Information and Software Technology*, vol. 92, pp. 1–14, 2017, doi: 10.1016/j.infsof.2017.07.001.

[22] S. Misra, V. Kumar, and U. Kumar, “Identifying some important success factors in adopting agile software development practices,” *Journal of Systems and Software*, vol. 82, no. 11, pp. 1869–1880, 2009, doi: 10.1016/j.jss.2009.06.006.

[23] M. K. Sharma and S. Mishra, “Predictive analytics in software engineering: A systematic review,” *IEEE Access*, vol. 7, pp. 158915–158935, 2019, doi: 10.1109/ACCESS.2019.2949781.

[24] R. Kupiainen, M. V. Mäntylä, and J. Itkonen, “Using metrics in agile and lean software development – A systematic literature review,” *Information and Software Technology*, vol. 62, pp. 143–163, 2015, doi: 10.1016/j.infsof.2015.02.005.

[25] B. Fitzgerald and K. Stol, “Continuous software engineering: A roadmap and future directions,” *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017, doi: 10.1016/j.jss.2015.06.063.