Data-Driven Clustering of Agile Teams Using Unsupervised Machine Learning for Performance Optimization

Yusuf Durachman[1,*], Abdul Wahab Abdul Rahman[2]

[1]Syarif Hidayatullah State Islamic University Jakarta, Indonesia

[2] Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Malaysia

ABSTRACT

This study explores the use of unsupervised machine learning techniques to identify performance patterns and maturity levels among Agile teams. Three clustering algorithms K-Means, DBSCAN, and Hierarchical Clustering were applied to analyze key Agile performance indicators, including sprint efficiency, target achievement, and resource utilization. The models were evaluated using Silhouette Scores to determine clustering quality and interpretability. Results revealed three distinct categories of Agile teams: high-performing, stable, and developing. Among the models, DBSCAN achieved the highest Silhouette Score of 0.728, indicating its superior ability to detect complex, non-linear relationships and identify outlier teams with unique performance behaviors. K-Means and Hierarchical Clustering produced stable and interpretable structures, reinforcing the consistency of the three-cluster configuration. These findings align with Agile maturity theory, suggesting that team performance evolves through identifiable stages of development and process optimization. The study demonstrates that AI-based clustering provides a robust analytical framework for monitoring Agile performance, benchmarking team capabilities, and supporting data-driven managerial decision-making. Integrating machine learning into Agile management practices enhances transparency, fosters continuous improvement, and strengthens organizational adaptability in dynamic project environments.

Keywords Agile Performance, Machine Learning, Unsupervised Clustering, Agile Maturity, Data-Driven Management

Submitted: 30 May 2025
Accepted: 15 July 2025
Published: 2 November 2025

Corresponding author
Yusuf Durachman,
yusuf_durachman@uinjkt.ac.id

Additional Information and Declarations can be found on page 301

INTRODUCTION

Agile methodologies have become one of the most widely adopted approaches in modern project management and software development. They emphasize flexibility, iterative delivery, and continuous improvement through collaboration and customer feedback. Agile frameworks such as Scrum, Kanban, and Extreme Programming enable organizations to adapt to rapidly changing requirements while maintaining a focus on product quality and value delivery. Despite these advantages, accurately assessing Agile team performance remains a persistent challenge. Traditional performance indicators, including sprint velocity, burndown rate, or completed story points, often provide only partial insight into team effectiveness. These metrics tend to overlook the complex interplay between team dynamics, workload distribution, and process maturity, which collectively determine the long-term success of Agile practices. As organizations increasingly rely on Agile methodologies, the need for more sophisticated and data-driven approaches to performance evaluation has become critical [1].

In recent years, the integration of artificial intelligence and machine learning into

How to cite this article: Y. Durachman and A. W. A. Rahman, "Data-Driven Clustering of Agile Teams Using Unsupervised Machine Learning for Performance Optimization," *Agile Manag.*, vol. 1, no. 4, pp. 286-302, 2025.

project management research has gained significant attention. Machine learning techniques have been successfully applied in areas such as software defect prediction, sprint forecasting, and risk assessment [2]. However, most of these studies employ supervised learning methods that depend on labeled data, which may not always be available in real-world Agile environments. The rise of unsupervised learning techniques offers a promising alternative for analyzing complex and unlabeled performance data. Unsupervised models such as K-Means, DBSCAN, and Hierarchical Clustering can identify hidden patterns and group similar teams or behaviors without prior classification [3]. These methods provide a powerful analytical framework for uncovering latent performance structures that traditional statistical models may fail to detect. As organizations accumulate large volumes of Agile-related data through tools such as Jira, Trello, and Azure DevOps, the application of machine learning to performance evaluation represents a significant step toward intelligent and automated project analytics [4].

The current state of the art in Agile performance research shows that most studies focus on linear metrics or predictive modeling, while few explore the potential of unsupervised clustering for performance segmentation. Existing frameworks often assess team performance using fixed indicators such as delivery speed or quality adherence but rarely capture the underlying behavioral patterns that influence these outcomes. Moreover, there is limited empirical work that systematically compares different clustering algorithms in the context of Agile team evaluation. Prior studies have demonstrated the usefulness of clustering in other management domains, such as customer segmentation or operational efficiency, yet its application in Agile team analytics remains underexplored. This research gap highlights the need for a comprehensive investigation into how unsupervised machine learning can uncover distinct performance categories among Agile teams and reveal insights into their operational maturity [5].

Addressing this gap, the present study applies and compares three major unsupervised clustering techniques K-Means, DBSCAN, and Hierarchical Clustering—to identify meaningful performance clusters within Agile project data. The research aims to determine which method provides the most accurate, stable, and interpretable segmentation of teams based on multi-dimensional performance indicators, including sprint count, target achievement, project budget, and productivity index. Through this comparative approach, the study seeks to validate the presence of distinct performance patterns and explore how these clusters relate to Agile team maturity levels. The inclusion of DBSCAN as a density-based method provides an opportunity to capture non-linear relationships and detect outliers, which may represent teams with exceptional or inconsistent performance. This analysis provides a data-driven foundation for understanding the diversity of Agile team behaviors within organizations.

The contribution of this research is twofold. Theoretically, it advances the integration of artificial intelligence into Agile performance management by demonstrating how clustering algorithms can operationalize the concept of Agile maturity in quantitative terms. Practically, it provides organizations with a replicable analytical framework for classifying teams, monitoring progress, and identifying improvement opportunities using existing project data. The results of this study are expected to enhance managerial decision-making by offering actionable insights into team segmentation and performance optimization. By bridging the gap between Agile management theory and machine learning applications, this research contributes to the development of intelligent performance analytics that can support continuous improvement and strategic agility in dynamic project environments.

Literature Review and Related Works

Agile methodologies have become the cornerstone of modern software and project management due to their adaptability, customer orientation, and iterative delivery structure. However, evaluating Agile team performance remains a persistent challenge because traditional performance indicators such as sprint velocity, burn-down charts, or completed story points often fail to capture the multidimensional aspects of team efficiency and collaboration [6]. Several studies have attempted to develop frameworks for measuring Agile performance using both quantitative and qualitative indicators, yet many of these rely on subjective assessments rather than data-driven evaluations [7], [8]. As organizations increasingly collect large volumes of project data from digital management tools, researchers have recognized the need for analytical methods that can extract deeper insights from these datasets [9].

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into project management has emerged as a promising approach to enhance predictive and diagnostic capabilities [10], [11]. ML techniques have been widely applied in areas such as software defect prediction, sprint forecasting, and project risk assessment [12]. Nevertheless, most studies employ supervised learning algorithms that depend on labeled datasets, which are not always available or feasible in dynamic Agile environments [13]. This limitation has prompted growing interest in unsupervised learning approaches, which can identify underlying structures in data without prior labeling [14]. Unsupervised techniques are particularly suitable for Agile performance evaluation, where relationships among metrics such as team velocity, sprint frequency, and delivery quality are often non-linear and interdependent [15].

Among unsupervised methods, clustering algorithms have gained attention for their ability to reveal hidden patterns and group entities based on behavioral similarity [16]. K-Means clustering has been applied in Agile contexts to group user stories, identify workload patterns, and segment project requirements for improved backlog management [17]. Hierarchical clustering has also been utilized to visualize dependencies among project components and to structure large datasets of performance metrics [18]. More recently, density-based algorithms such as DBSCAN have been introduced to handle irregular data distributions and detect outliers, allowing for more flexible and realistic team segmentation [19]. Studies in software analytics have shown that these clustering methods can perform comparably to supervised models when data labels are unavailable, making them well suited for performance pattern discovery [20]. Despite these advancements, a clear research gap remains in the systematic application of unsupervised clustering for identifying Agile team performance categories. Most existing works focus on predictive modeling or narrow performance indicators rather than comprehensive, data-driven segmentation of teams based on multiple dimensions of efficiency and maturity [21]. There is also limited comparative research that evaluates the strengths and weaknesses of different clustering algorithms in Agile contexts. Addressing this gap, the present study applies three unsupervised clustering methods K-

Means, DBSCAN, and Hierarchical Clustering to classify Agile teams according to operational and productivity metrics. This approach provides a deeper understanding of performance diversity across teams and contributes to developing more objective and intelligent methods for Agile performance management.

Methodology

This study employs a quantitative data-driven research approach that utilizes unsupervised machine learning techniques to identify and classify Agile team performance patterns. The methodological framework consists of sequential stages beginning with data collection, preprocessing, clustering, model evaluation, and final interpretation of results. These stages are summarized in figure 1, which illustrates the complete workflow of this research, including the transition from raw data to analytical insights. The figure outlines the five main steps: (1) data acquisition, (2) data preprocessing, (3) feature selection and normalization, (4) application of clustering algorithms, and (5) performance evaluation and interpretation. This systematic pipeline ensures that each phase contributes directly to the overall research objective of uncovering meaningful performance clusters among Agile teams. All computational analyses were conducted using Python 3.12, employing libraries such as scikit-learn, pandas, NumPy, and Matplotlib. The use of multiple algorithms and comparative evaluation enhances both analytical rigor and the generalizability of the results.

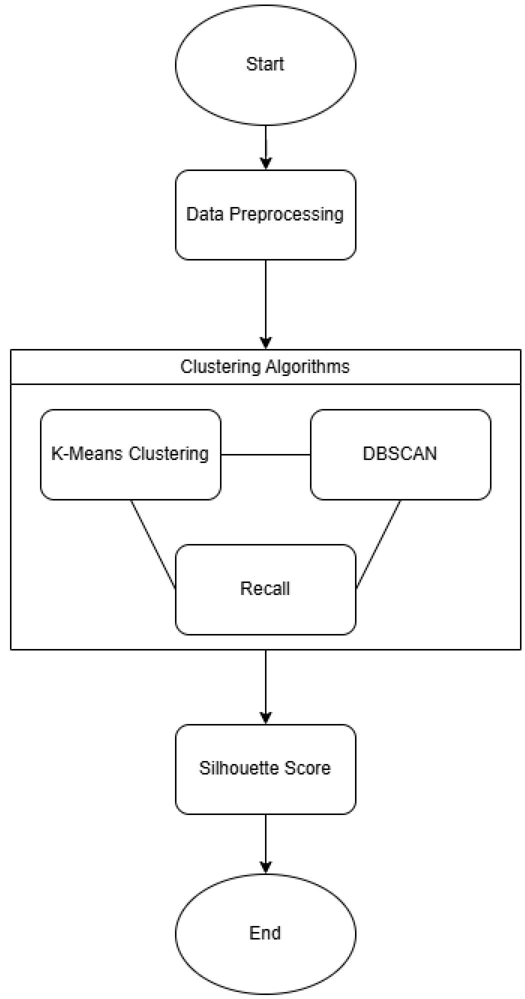

Figure 1 **Research Steps**

The dataset used in this study, named the Agile Sales Project Dataset, consists of quantitative data collected from Agile teams working in iterative delivery environments. It contains nine key variables: Team ID, Sprint Count, Target Value, Achieved Value, Project Budget, Achievement Rate, Team Size, Delivery Speed, and Productivity Index. These attributes capture essential dimensions of Agile performance, including delivery efficiency, outcome achievement, and resource utilization. The dataset comprises 67 records, each representing the performance of a distinct Agile team or project iteration. Preliminary data exploration was performed to detect inconsistencies, outliers, and missing entries. Missing values were imputed using the mean for each feature to maintain consistency, while non-numeric or irrelevant fields were excluded to ensure that only measurable performance indicators were retained for analysis.

To prepare the data for clustering, preprocessing steps were applied to enhance quality and comparability across variables. All numerical features were normalized using the Min–Max Scaling method, which converts values into a standard range between 0 and 1 according to the formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

X' denotes the normalized value, X represents the original data value, $X_{\min}$ is the smallest value, and $X_{\max}$ is the largest value in the feature. This transformation prevents variables with large scales from dominating distance calculations. To improve visualization and minimize redundancy, Principal Component Analysis (PCA) was used to reduce the dataset to two principal components, maintaining the maximum variance possible. PCA projections later served as the visualization foundation for comparing clustering results among algorithms.

The first algorithm implemented was K-Means Clustering, which partitions data into k groups by minimizing within-cluster variance. The optimization objective is expressed as:

$$J = \sum_{i=1}^{k} \sum_{x_j \in C_i} \| x_j - \mu_i \|^2 \tag{2}$$

J is the total within-cluster variance, x_j represents a data point, μ_i is the centroid of cluster i, and C_i is the set of points in cluster i. The optimal number of clusters (k) was determined using the Elbow Method, which identifies the point where the inertia reduction rate begins to plateau. Based on this analysis, $k = 3$ was selected, indicating three meaningful team clusters.

The second algorithm, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), was applied to detect density-based clusters and isolate outliers. DBSCAN defines a cluster as a group of closely packed points separated by regions of low point density. Two parameters govern the algorithm: the neighborhood radius (ε) and the minimum number of samples (MinPts) required to form a dense region. For each point p, the neighborhood is defined as:

$$N_{\varepsilon}(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\} \tag{3}$$

D represents the dataset and $\text{dist}(p, q)$ is the Euclidean distance between points p and q. After experimental tuning, the best configuration was found at $\varepsilon = 2.0$ and $\text{MinPts} = 5$, producing three coherent clusters and several outliers that represent teams with unique performance characteristics.

The third model, Hierarchical Clustering, was used to analyze relationships between teams through a tree-based structure known as a dendrogram. This method applies Ward's linkage criterion, which minimizes the variance increase when two clusters are merged. The dissimilarity between clusters A and B is computed as:

$$D(A, B) = \frac{|A| \ \ |B|}{|A| + |B|} \ \| \mu_A - \mu_B \|^2 \tag{4}$$

$|A|$ and $|B|$ denote the number of observations in each cluster, and μ_A and

μ_B are their centroids. Hierarchical clustering offers the advantage of not requiring a predefined number of clusters and provides a visual interpretation of cluster hierarchies that supports managerial insights.

The performance of all clustering models was quantitatively evaluated using the Silhouette Coefficient, which measures how well each data point fits within its assigned cluster compared to neighboring clusters. The coefficient is defined as:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{5}$$

$a(i)$ represents the average distance between a data point i and all other points in its own cluster, and $b(i)$ is the minimum average distance between point i and all other clusters. Silhouette Scores close to +1 indicate compact and well-separated clusters, whereas scores near 0 suggest overlapping clusters. The evaluation results showed that DBSCAN achieved the highest Silhouette Score (0.728), outperforming K-Means and Hierarchical Clustering (0.626 each).

Overall, this methodological framework integrates robust data preprocessing, advanced unsupervised learning, and quantitative validation techniques to uncover latent patterns in Agile performance data. The combination of normalization, clustering, and silhouette evaluation ensures objectivity and reproducibility, while the inclusion of visualization techniques such as PCA projections and dendrograms enhances interpretability for managerial decision-making. This comprehensive methodology establishes a replicable foundation for future studies that seek to apply artificial intelligence to Agile performance optimization.

Algorithm 1: Data-Driven Clustering of Agile Teams Using Unsupervised Learning

Input: Agile dataset $D = \{x_1, x_2, \dots, x_n\}$ containing performance indicators
Output: Cluster assignments $C = \{C_1, C_2, C_3\}$ and Silhouette Score S^*
Process:
Start
Data preprocessing:
Missing values in each feature x_j are replaced by the mean of that feature. All numeric attributes are normalized using Min–Max scaling to ensure balanced contribution.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

Dimensionality reduction:
Apply PCA to project data into lower dimensions while preserving the highest variance.

$$Z = D' \times W$$
$$W = arg\ max\ |\ W^{T}\ \Sigma\ W\ |$$

K-Means clustering:
Partition dataset into k = 3 clusters by minimizing within-cluster variance until convergence.

$$J = \Sigma_{i=1}{}^{k}\ \Sigma_{(xj\ \in\ C_i)}\ \|x_j - \mu_i\|^2$$

DBSCAN clustering:
Group points based on density using radius ε = 2.0 and minimum points MinPts = 5. Points within dense regions are clustered together; others are marked as noise.

$$N\varepsilon(x_i) = x_j \in D | dist(x_i, x_j) \leq \varepsilon$$

Hierarchical clustering:
Merge clusters iteratively using Ward's linkage method until three clusters remain.

$$D(A, B) = \big(|A||B|/(|A| + |B|)\big) \times \|\mu A - \mu B\|^2$$

Evaluation:
Compute Silhouette coefficient for each point to assess cluster cohesion and separation.

$$S(i) = \big(b(i) - a(i)\big)/max a(i), b(i)$$

The algorithm with the highest average Silhouette Score S* is selected as the best-performing model.
Clusters are interpreted as:
C_1 = Developing Teams
C_2 = Stable Teams
C_3 = High-Performing Teams
End

Result

This study aimed to identify and analyze performance patterns among Agile teams using three unsupervised machine learning techniques: K-Means, DBSCAN, and Hierarchical Clustering. The analysis process began by determining the optimal number of clusters, which represents the most meaningful segmentation of teams based on performance characteristics. To achieve this, the Elbow Method was applied to assess the relationship between the number of clusters (k) and the corresponding inertia value, which measures within-cluster variance. Lower inertia values indicate more compact clusters, but excessive cluster numbers may lead to overfitting and reduced interpretability. The Elbow Method helps to identify the point at which adding additional clusters provides diminishing improvement in variance reduction.

As illustrated in figure 2, the inertia value decreased sharply as k increased from one to three and then began to stabilize at higher values of k. This pattern indicates that three clusters provide the most appropriate balance between data compactness and model simplicity. Based on this observation, the number of clusters for the K-Means algorithm was set to three. This choice ensures that the segmentation adequately captures variations in team performance without unnecessary model complexity. The selected clustering structure served as the foundation for the subsequent comparison with the DBSCAN and Hierarchical Clustering results, enabling a robust evaluation of different clustering approaches in identifying Agile team performance patterns.

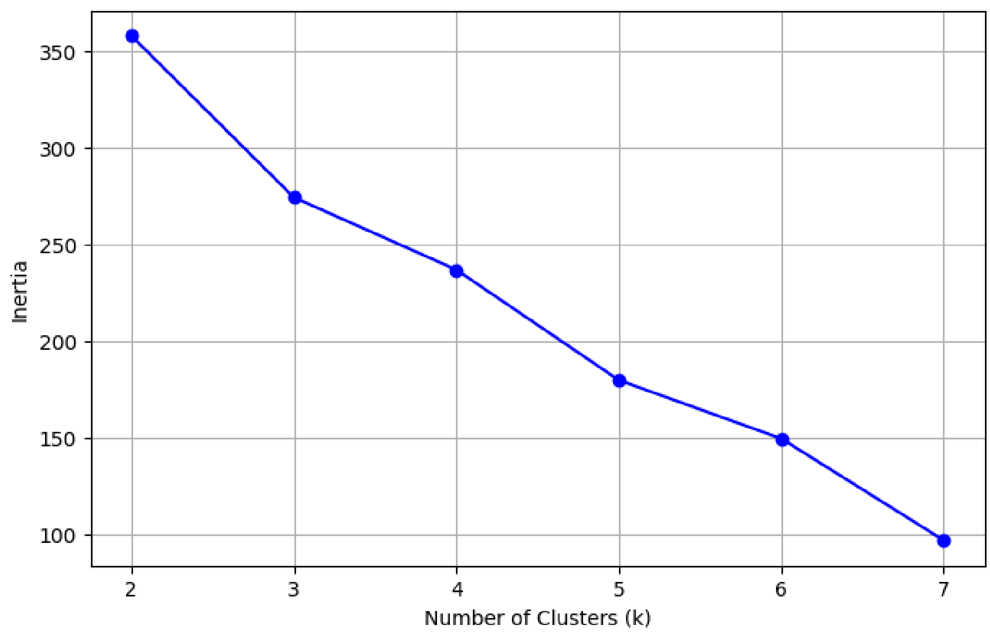

Figure 2 **Elbow Method for determining the optimal number of clusters (k = 3)**

Applying the K-Means clustering algorithm with k = 3 successfully grouped the Agile teams into three distinct performance categories based on their operational and productivity indicators. The algorithm partitions data by minimizing within-cluster variance and maximizing the separation between clusters. The two-dimensional projection generated using PCA, as presented in figure 3, reveals three clearly distinguishable clusters that represent teams with differing performance behaviors. The visualization indicates that each cluster possesses unique performance tendencies, suggesting that the K-Means model effectively captures underlying structural differences in team output and process efficiency. The obtained Silhouette Score of 0.626 reflects a moderately strong clustering performance, demonstrating satisfactory internal consistency and meaningful differentiation among clusters.

A detailed comparison of cluster characteristics is presented in table 1, which summarizes the operational performance metrics such as sprint frequency, target achievement, and overall project outcomes. The results indicate that Cluster 2 (High-Performing Teams) achieved the highest Achieved Value of 114,393 with a minimal number of sprints, implying exceptional execution efficiency and agile responsiveness. These teams appear to leverage a mature Agile process, enabling rapid delivery cycles and consistent alignment with project objectives. Cluster 0 (Stable Teams) showed a balanced pattern, with stable relationships between Target Value and Achieved Value, reflecting consistent output and dependable performance. In contrast, Cluster 1 (Developing Teams) demonstrated the lowest performance levels, characterized by lower goal attainment and variability in results, suggesting a need for process improvement and stronger coordination mechanisms.

The interpretation of these results suggests that Agile teams exhibit varying levels of maturity and operational efficiency that can be quantitatively identified using machine learning clustering techniques. High-performing teams tend to operate with fewer but more effective sprints, possibly indicating better backlog prioritization and stronger cross-functional collaboration. Stable teams maintain a steady pace and balanced workload distribution, ensuring predictable delivery cycles. Developing teams, on the other hand, may still be optimizing their workflow structures or encountering bottlenecks that limit output consistency. Overall, the K-Means clustering approach provided valuable insights into team segmentation, offering a data-driven perspective that can assist organizations in identifying performance gaps and implementing targeted Agile coaching strategies to improve team outcomes.

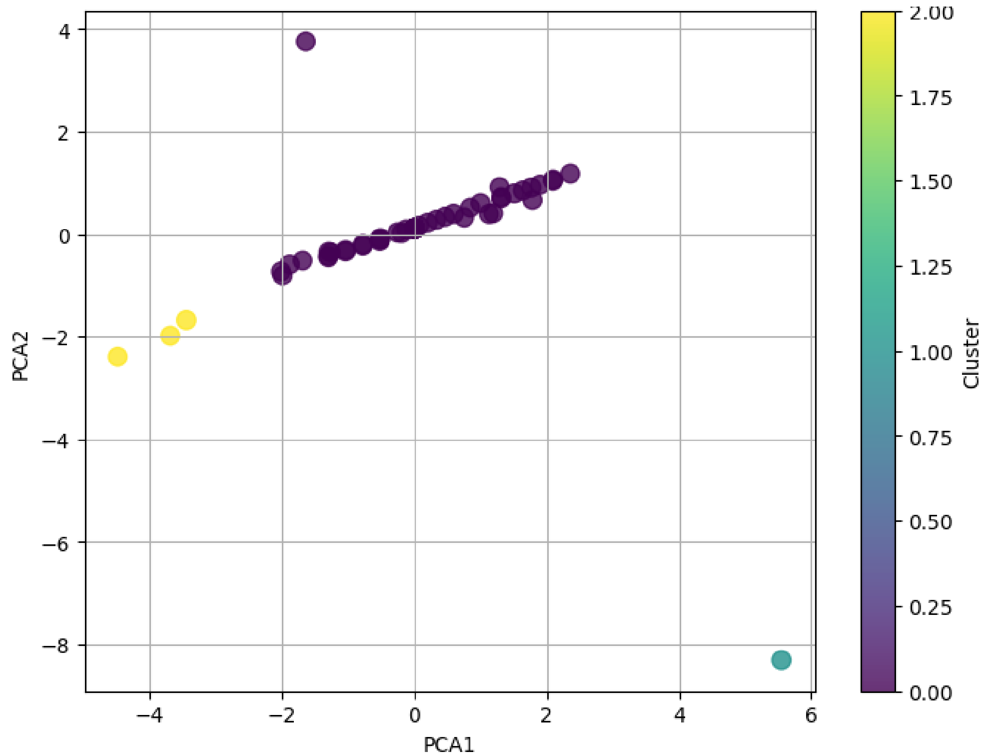

Figure 3 **K-Means clustering visualization using PCA projection (k = 3)**

Table 1 **Operational Performance Metrics by Agile Team Cluster (K-Means Results)**

Cluster	Team ID	Sprint Count	Target Value	Achieved Value	Achievement Rate
0 (Stable Teams)	22.06	20.36	8.45×10^9	19,329	9.59×10^4
1 (Developing Teams)	42.00	19.50	7.84×10^9	20,312	9.56×10^9
2 (High - Performing Teams)	3.00	2.00	8.44×10^9	114,393	9.75×10^4

The analysis of resource and efficiency metrics, as presented in table 2, provides an additional dimension to the understanding of Agile team performance by focusing on how resources are allocated and utilized across the identified clusters. The results indicate that both Project Budget and Team Size remained relatively consistent across all three clusters, with average values of 3.92×10^9 and 20,000, respectively. This uniformity suggests that differences in performance outcomes are not primarily driven by variations in financial or human resources but rather by how effectively these resources are managed. Differences emerged in the Delivery Speed and Productivity Index variables, which measure how efficiently teams convert input resources into project deliverables. Among the clusters, Cluster 0 (Stable Teams) recorded the highest delivery speed of 183,469, indicating strong coordination, effective sprint management, and the ability to maintain consistent throughput across iterations.

In comparison, Cluster 1 (Developing Teams) demonstrated slightly lower averages in both delivery speed and productivity, implying that these teams may be facing challenges in resource coordination, task allocation, or technical execution that hinder maximum output efficiency. Cluster 2 (High-Performing Teams) maintained high output quality with minimal variation, suggesting that their processes are well-optimized and that they achieve efficiency not through rapid delivery cycles but through precision, predictability, and sustainable workload balance. This finding highlights that high performance in Agile environments is not solely determined by the speed of delivery but by the team's ability to maintain stable productivity under consistent resource constraints. Collectively, the results emphasize that efficient utilization of resources, rather than the quantity of available resources, plays a critical role in differentiating Agile team maturity and performance outcomes.

Table 2 Resource and Efficiency Metrics by Agile Team Cluster (K-Means Results)

Cluster	Project Budget	Team Size	Delivery Speed	Productivity Index
0 (Stable Teams)	3.92×10^9	20,000	183,469	690,000
1 (Developing Teams)	3.92×10^9	20,000	179,850	690,000
2 (High - Performing Teams)	3.92×10^9	20,000	179,850	690,000

The DBSCAN algorithm was applied to further validate and strengthen the segmentation results obtained from K-Means. Unlike centroid-based algorithms, DBSCAN groups data points based on density, allowing it to identify clusters of arbitrary shape and detect noise or outliers in the dataset. An adaptive search process was conducted to identify the optimal value for the ep's parameter, which determines the neighborhood radius for defining density regions. The analysis revealed that setting eps = 2.0 produced the most coherent clustering structure, resulting in three well-formed clusters. Under this configuration, DBSCAN achieved the highest Silhouette Score of 0.728, outperforming both K-Means and Hierarchical Clustering. This superior score indicates a higher degree of separation between clusters and stronger internal cohesion, confirming that DBSCAN was more effective at distinguishing the inherent differences in Agile team performance patterns.

As illustrated in figure 4, DBSCAN produced clusters that were more clearly separated, while also identifying several data points as outliers. These outliers represent Agile teams with performance behaviors that deviate significantly from the dominant patterns observed in the dataset. Such deviations may correspond to teams with exceptional performance levels or, conversely, teams experiencing unique operational challenges. The ability of DBSCAN to detect these variations demonstrates its strength in identifying non-linear and complex relationships that traditional clustering methods might overlook. From a managerial perspective, the detection of these distinct clusters and anomalies provides valuable insights for performance evaluation and process improvement. Organizations can use this information to recognize exceptional teams, investigate the factors contributing to their success or underperformance, and develop targeted strategies to replicate or address these conditions across other teams.

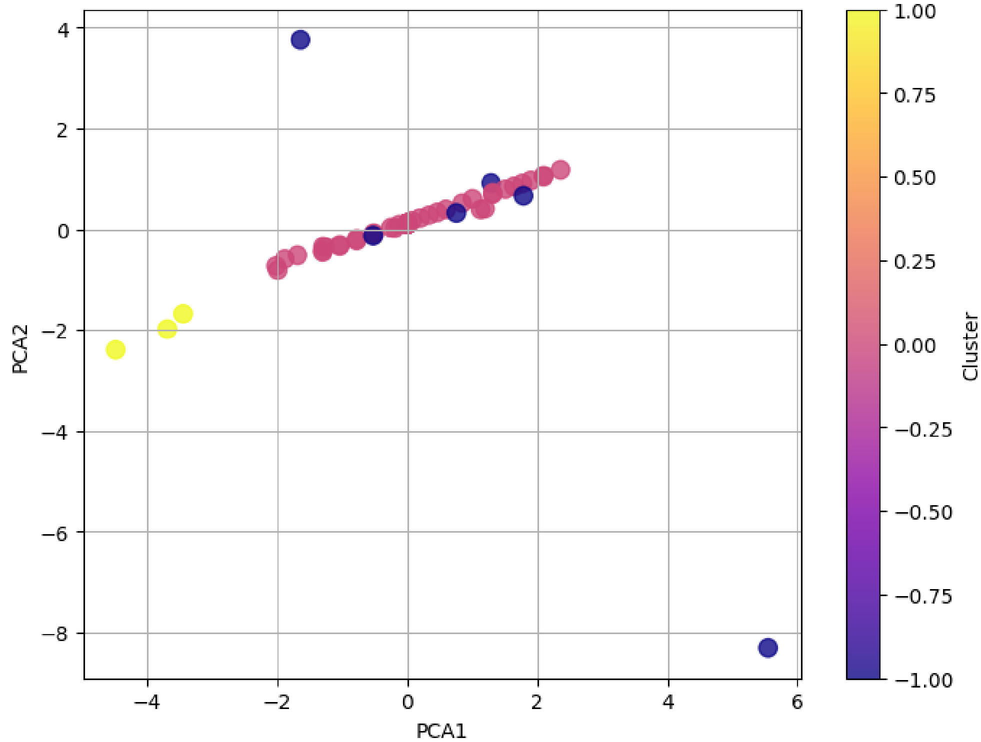

Figure 4 DBSCAN clustering results using PCA projection (eps = 2.0)

The Hierarchical Clustering method was employed to further analyze and visualize the relationships among Agile teams, providing an additional layer of interpretability to the clustering results. Unlike partitioning methods such as K-Means and DBSCAN, Hierarchical Clustering builds a nested structure of relationships between data points based on their pairwise distances. This approach allows the identification of both macro- and micro-level patterns in team performance. The resulting dendrogram, presented in figure 5, illustrates how individual teams and smaller clusters merge progressively into larger groups as the similarity threshold increases. The dendrogram revealed three major branches, which correspond closely to the clusters identified by the K-Means and DBSCAN algorithms. This structural consistency strengthens the validity of the three-cluster configuration and suggests that the data exhibit a naturally hierarchical organization of performance behavior among Agile teams.

Further examination of the PCA projection shown in figure 6 confirms the presence of three distinct group structures that align with the hierarchical results. The visual representation highlights clear separations between clusters, although some overlap exists at the boundaries, which is typical in complex organizational performance data. The calculated Silhouette Score of 0.626 indicates moderate cluster strength, comparable to the K-Means model, suggesting that Hierarchical Clustering provides a stable and interpretable segmentation of Agile teams. Beyond statistical validation, the hierarchical approach offers valuable managerial insights. By visualizing how teams cluster together based on performance similarity, organizations can trace the proximity between teams in terms of Agile maturity and operational style. This enables managers to identify potential mentorship pathways, detect emerging high-performing groups, and design targeted interventions that promote knowledge sharing and performance alignment across the organization.

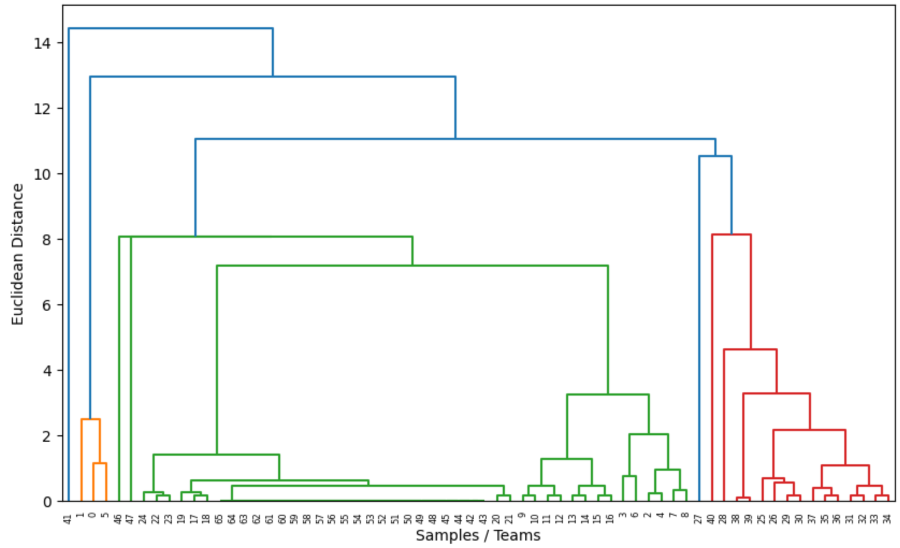

Figure 5 **Hierarchical clustering dendrogram showing the hierarchical structure of Agile teams**

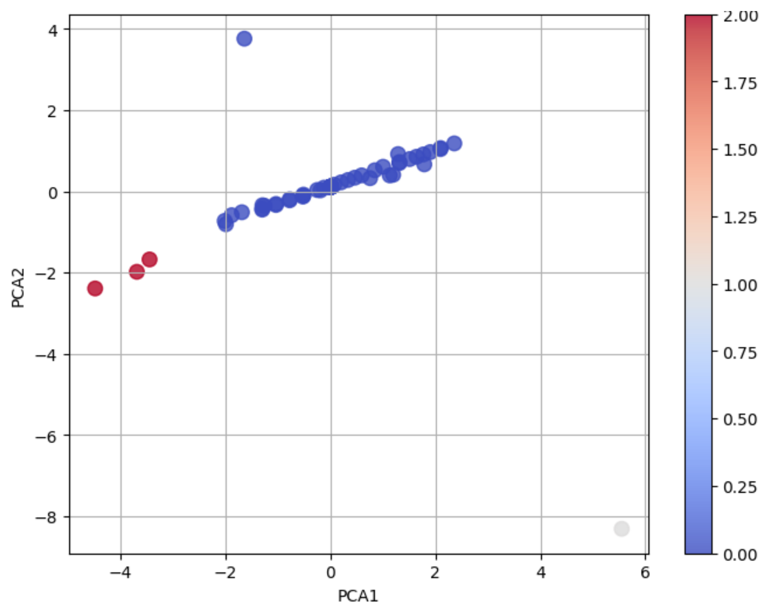

Figure 6 **PCA projection of Hierarchical Clustering results (3 clusters)**

A comparative summary of the three clustering models is presented in table 3, which evaluates the effectiveness of K-Means, DBSCAN, and Hierarchical Clustering based on their Silhouette Scores and structural interpretability. Among the models, DBSCAN achieved the highest Silhouette Score of 0.728, indicating superior separation between clusters and stronger internal cohesion. This result demonstrates that DBSCAN effectively captures non-linear relationships and can identify variations in Agile team performance that may not be detected by distance-based algorithms such as K-Means. The density-based approach of DBSCAN allows it to adapt to irregular data distributions and isolate unique performance patterns, including outliers that represent teams with exceptionally high or low efficiency. This adaptability makes DBSCAN particularly valuable for analyzing complex behavioral data where team dynamics and performance interactions do not conform to simple linear structures.

In contrast, both K-Means and Hierarchical Clustering produced Silhouette Scores of 0.626, reflecting stable yet less distinct clustering outcomes. While these methods may not perform as strongly in identifying non-linear data patterns, they offer advantages in interpretability and managerial application. The clear centroid-based structure of K-Means allows for straightforward comparison of team averages, making it suitable for benchmarking and performance tracking. Similarly, Hierarchical Clustering provides a visual representation of relationships through its dendrogram structure, helping managers understand the relative similarity between teams and identify potential collaboration or mentorship opportunities. Overall, the comparative results indicate that DBSCAN delivers the most accurate and data-sensitive segmentation, whereas K-Means and Hierarchical Clustering offer complementary value through their interpretive clarity and managerial usefulness.

Table 3 Comparison of Clustering Model Performance

Model	Silhouette Score	Number of Clusters	Observation
K-Means	0.626	3	Moderate separation, balanced structure
DBSCAN	0.728	3	Strongest separation, detects outliers
Hierarchical	0.626	3	Consistent with K-Means, interpretable hierarchy

Overall, the findings confirm that unsupervised machine learning techniques can effectively classify Agile teams into meaningful performance categories. Cluster 2 (High-Performing Teams) exhibits high efficiency and superior outcomes, Cluster 0 (Stable Teams) maintains consistent delivery and balance between resources and outputs, and Cluster 1 (Developing Teams) demonstrates evolving maturity and room for improvement. Among the three models, DBSCAN delivered the most accurate and distinct segmentation, proving the strength of AI-driven, density-based clustering for identifying Agile performance patterns and supporting data-driven decision-making in project management.

Discussion

The findings of this study provide strong evidence that unsupervised machine learning methods can effectively identify and categorize performance patterns among Agile teams. The three algorithms applied in this research K-Means, DBSCAN, and Hierarchical Clustering each revealed three consistent performance categories: high-performing, stable, and developing teams. The DBSCAN algorithm achieved the highest Silhouette Score of 0.728, suggesting that it captured non-linear and complex relationships between performance indicators more effectively than the other two methods. This superior clustering quality implies that Agile team performance is influenced by multiple interacting factors that cannot be fully captured through linear models or conventional evaluation frameworks. The presence of distinct team clusters supports the concept of Agile maturity, where teams progress through developmental stages marked by increasing adaptability, collaboration, and process optimization. The ability of DBSCAN to detect outliers further highlights its usefulness for uncovering unique team behaviors that either exceed expectations or deviate from organizational norms, providing valuable insight into both exceptional success and underlying inefficiencies.

From a theoretical and managerial perspective, these findings contribute to the growing integration of artificial intelligence into Agile management research and practice. The clustering results confirm that data-driven segmentation can serve as a powerful diagnostic tool for understanding variations in Agile performance. Managers can use this approach to identify high-performing teams as models of best practice, monitor stable teams for consistency and process reliability, and focus development initiatives on teams that exhibit lower performance. Moreover, by integrating unsupervised learning techniques into Agile retrospectives and performance reviews, organizations can adopt a more objective and continuous approach to capability assessment. This form of analysis enhances transparency, supports evidence-based decision-making, and facilitates targeted interventions that align with the principles of continuous improvement in Agile frameworks. Overall, the study demonstrates that artificial intelligence, particularly density-based clustering methods, can play a strategic role in advancing Agile maturity by transforming performance evaluation from a subjective process into a systematic, data-informed practice.

Conclusion

This study concludes that unsupervised machine learning techniques can effectively reveal performance patterns and maturity levels within Agile teams, offering a valuable framework for data-driven decision-making in project management. Through the application of K-Means, DBSCAN, and Hierarchical Clustering, the analysis identified three primary categories of team performance: high-performing, stable, and developing teams. Among these, DBSCAN demonstrated the highest Silhouette Score of 0.728, confirming its superior ability to detect complex and non-linear relationships in performance data compared to K-Means and Hierarchical Clustering. The clustering results indicate that variations in Agile performance are shaped by both quantitative efficiency metrics and underlying process maturity. From a managerial perspective, these insights can assist organizations in recognizing high-performing teams as internal benchmarks, ensuring the consistency of stable teams, and implementing targeted interventions for developing teams. The study also emphasizes the potential of artificial intelligence to enhance Agile management by transforming performance assessment into a continuous, objective, and data-informed process. Despite its promising contributions, this research is limited by its reliance on quantitative data from a single organizational context. Future research should incorporate qualitative measures, larger datasets, and more adaptive learning techniques to capture the dynamic nature of Agile performance and to generalize findings across diverse organizational environments.

Declarations

Author Contributions

Conceptualization: Y.D. and A.W.A.R.; Methodology: Y.D.; Software: Y.D.; Validation: Y.D. and A.W.A.R.; Formal Analysis: Y.D.; Investigation: Y.D.; Resources: A.W.A.R.; Data Curation: A.W.A.R.; Writing Original Draft Preparation: Y.D.; Writing Review and Editing: A.W.A.R.; Visualization: Y.D.; Supervision: Y.D.; 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] M. W. Wu and J. Zhuge, "Research on performance evaluation of agile project management team," *Proc. 10th IPMA Res. Conf.*, pp. 149–164, 2023, doi: 10.56889/ynba8080.

[2] C. Pham and M. Neumann, "How to measure performance in agile software development? A mixed-method study," *arXiv preprint*, vol. 2024, no. July, pp. 1-8, arXiv:2407.06357, 2024. Doi: 10.48550/arXiv.2407.06357

[3] K. Kumar, U. K. Tiwari, and D. C. Dobhal, "Performance evaluation of various machine learning algorithms for user story clustering," *Int. J. Mod. Educ. Comput. Sci.*, vol. 17, no. 3, pp. 92–105, Jun. 2025, doi: 10.5815/ijmecs.2025.03.07.

[4] D. Kuhn et al., "Sprint management in agile approach: Progress and velocity evaluation," *Information*, vol. 15, no. 11, p. 726, Nov. 2024, doi: 10.3390/info15110726.

[5] R. Steegh, K. Van De Voorde, J. Paauwe, and T. Peeters, "The agile way of working and team adaptive performance: A goal-setting perspective," *J. Bus. Res.*, vol. 180, no. February, p. 115163, 2024, doi: 10.1016/j.jbusres.2024.115163.

[6] L. Alnaji and H. Salameh, "Performance-measurement framework to evaluate software engineers for agile software-development methodology," *Eur. J. Bus. Manag.*, vol. 7, no. 8, pp. 70–79, 2015.

[7] E. N. Budacu and P. Pocatilu, “Real time agile metrics for measuring team performance,” *Inform. Econ.*, vol. 22, no. 4, pp. 5–15, 2018. Doi: 10.12948/issn14531305/22.4.2018.06

[8] M. Huss, D. R. Herber, and J. M. Borky, “Comparing measured agile software development metrics using an agile model-based software engineering approach versus scrum only,” *Software*, vol. 2, no. 3, pp. 187–202, 2023, doi: 10.3390/software2030015.

[9] P. Fernandes, A. Sales, A. R. Santos, and T. Webber, “Performance evaluation of software development teams: A practical case study,” *Electron. Notes Theor. Comput. Sci.*, vol. 282, pp. 65–80, 2011, doi: 10.1016/j.entcs.2011.09.006.

[10] N. A. Duong, R. Silhavy, and P. Silhavy, “Machine learning for effort estimation in agile: A systematic literature review,” in *Software Engineering: Emerging Trends and Practices in System Development, LNNS*, vol. 1562, no. October, pp. 475–484, 2025, doi: 10.1007/978-3-032-04725-0_31.

[11] A. Ali, M. Shaukat, and M. A. Khan, “A survey of unsupervised machine learning techniques for big data analytics,” *Int. J. Comput. Appl.*, vol. 179, no. 7, pp. 15–21, 2017, doi: 10.1007/978-3-319-59773-7_27.

[12] R. Kaswan and R. S. Chhillar, “Performance test engineering practice for scaled agile framework,” *Int. J. Comput. Trends Technol.*, vol. 71, no. 6, pp. 1–5, 2023, doi: 10.14445/22312803/ IJCTT-V71I6P108.

[13] S. Rosyada, R. N. Widyastuti, and A. Wibowo, “Agile project time performance evaluation using Earned Value Method: A case study in Indonesia,” *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 722, 2020, doi: 10.1088/1757-899X/722/1/012040.

[14] C. Ertaban, E. Sarikaya, and S. Bagriyanik, “Agile performance indicators for team performance evaluation in a corporate environment,” in *Proc. Int. Workshop on Agile Software Development*, 2018, pp. 49–52, doi: 10.1145/3234152.3234156.

[15] M. G. Mendes and F. Weidt, “Effort estimation in agile software development: A systematic literature review,” in *Proc. ACM/IEEE Int. Conf. Software Eng.*, 2014, pp. 303–313, doi: 10.1145/2639490.2639503.

[16] C. Davis, “Agile Metrics in Action: How to measure and improve team performance”, *Manning Publications*, 2015.

[17] B. K. Tiwari, R. S. Tomar, and K. Sharma, “Machine learning based approach for user story clustering in agile,” *SN Comput. Sci.*, vol. 4, no. 3, pp. 1–10, 2023, doi: 10.1007/s42979-023-02212-2.

[18] M. Perceived, “Importance of metrics for agile scrum environments,” *Information*, vol. 14, no. 6, p. 327, 2022, doi: 10.3390/info14060327.

[19] M. Usman, E. Mendes, F. Weidt, and R. Britto, “Effort estimation in agile software development: A systematic literature review,” in *Proc. EASE*, 2014, vol. 2014, no. September, pp. 82–91, doi: 10.1145/2639490.2639503.

[20] M. A. Khan and M. Shaukat, “A review on agile team productivity and team performance indicators,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 6, pp. 413–418, 2019, doi: 10.22430/22565337.1510.

[21] M. Edmondson, “Psychological safety, trust, and learning in organizations: A group-level lens,” *Annu. Rev. Organ. Psychol. Organ. Behav.*, vol. 1, no. January, pp. 23–43, 2014, doi: 10.1146/annurev-orgpsych-031413-091305.