Scientific Journal of Informatics
Vol. 13, No. 3, August 2026

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji e-ISSN 2460-0040

Automation of Sprint Cost Estimation in Agile Projects Using a Random Forest-Based Decision Support System

Faidatul Hikmah[1], Sinung Suakanto[2]
[1,2]Department of Information System, Telkom University, Indonesia

Abstract.
Purpose: The manual construction of a Work Breakdown Structure (WBS) is inherently susceptible to human cognitive biases; hence, the objective of this research shifts toward Intelligent Automation, leveraging a Decision Support System (DSS) to replicate expert financial logic.
Methods: The dataset comprises 30 validated sprint records from 23 internal projects; other WBS documents were excluded due to incomplete sprint details. Pre-processing involved completeness checks and One-Hot Encoding of team role compositions, without feature scaling. A Decision Tree Regressor, Random Forest Regressor, and Hybrid Model (Voting Regressor) were evaluated using 3-fold cross-validation based on MAPE, RMSE, and R^2 metrics.
Result: The Random Forest Regressor demonstrated the best relative performance among the three models, achieving a MAPE of 23.55%, an RMSE of IDR 35.27 million, and an R^2 of -0.535. Across the 30 sprint predictions, the Random Forest model showed a mean deviation of 23.55% ± 31.69%, a median of 9.98%, and a range of 0.78%–125.33%, highlighting performance variation across sprints and supporting the model's application within a human-in-the-loop framework. Feature importance analysis identified `sprint_duration_days` as the dominant feature, followed by `role_programmer_jr` and `role_qa_sr`.
Novelty: These findings demonstrate the system's practical contribution, particularly in streamlining budgeting bureaucracy by accelerating the process from days of manual preparation to just a few minutes through automated execution.

Keywords: Agile Sprint Cost, Decision Support System, Feature Importance, Software Cost Estimation, Random Forest Regressor

Received July 2026 / **Revised** August 2026 / **Accepted** August 2026

INTRODUCTION

In the modern software development industry, the Agile methodology has become a crucial approach due to its flexibility in iteratively responding to changing requirements. However, project economic governance continues to face challenges regarding software cost and effort estimation—particularly when estimates rely heavily on expert judgment, story points, or planning artifacts with varying information quality [1], [2], [3], [4]. In industrial practice, manually compiling cost components within a Work Breakdown Structure (WBS) demands a high level of consistency, while factors such as team experience, information quality, and iterative estimation processes can impact accuracy [1], [3], [5]. These obstacles underscore the need for computational tools that leverage historical data patterns to support decision-making without bypassing the Project Manager's validation.

Recent studies have applied data-driven cost/effort models, ensemble learning, Bayesian networks, tree-based models, natural language processing, and machine learning to Agile and software effort estimation [2], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. However, most research focuses on effort or story points at the project or user story level, utilizes cross-company public datasets, or employs functional attributes that do not align with internal Work Breakdown Structure (WBS) structures [6], [7], [8], [9], [10], [11], [12], [15], [16], [17], [18], [19]. The literature also indicates that differences in schemes, units of analysis, organizational contexts, and sample sizes can limit the direct transferability of models across datasets [20], [21], [22], [23], [24], [25], [26]. Consequently, the research gap addressed here

[1]*Corresponding author.
Email addresses: faiidamah@student.telkomuniverity.ac.id (Hikmah).
DOI: 10.15294/sji.v13i3.58315

lies not in the use of One-Hot Encoding or Random Forest as novel techniques, but rather in how operational sprint parameters specifically duration and role composition are translated into WBS cost structures at the sprint level of granularity and presented through a transparent Decision Support System (DSS).

To address this gap, this study presents a data-driven Decision Support System (DSS) based on supervised regression to replicate sprint-level WBS cost estimates. Human resource features are granularly decomposed using One-Hot Encoding, allowing the system to learn how variations in role composition across sprints interact with `sprint_duration_days`. The study aims to: (1) compare Decision Tree Regressor, Random Forest Regressor, and Hybrid Model (Voting Regressor) approaches using validated internal sprint data; (2) determine the final model using MAPE as the primary criterion, RMSE as the nominal error measure, and R^2 as a diagnostic metric; and (3) analyze feature importance to understand the non-linear relationships between sprint parameters and WBS cost estimates. The practical contribution is an initial estimation assistant that complements expert reviews and supports Project Manager decision-making [25], [26], [27], [28], [29], [30].

METHODS

The study employs an "Experiment-First" approach that integrates machine learning within the context of Software Engineering Economics. The research workflow is structured in stages, ranging from WBS data acquisition and validation, selection of complete sprint records, pre-processing and feature engineering, development of three regression models, 3-fold cross-validation, evaluation using MAPE/RMSE/R^2 metrics, and final model selection, to feature importance analysis. These stages are summarized in Figure 1 and detailed in the following section to ensure the process can be replicated with greater transparency [25], [26], [27], [28], [29], [30].

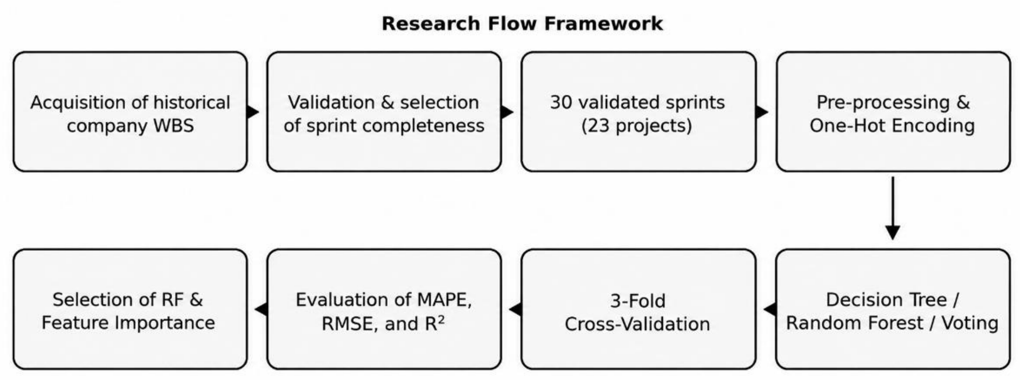

Figure 1. Research flow framework

The dataset used consists of primary, data-driven information sourced from historical Work Breakdown Structure (WBS) records of Agile-based software projects. Following the company's validation process, 30 sprint records from 23 projects met the criteria for sprint-level information completeness and were utilized as units of analysis, as summarized in Table 1. Other WBS documents were excluded due to insufficient detail regarding duration and/or role composition per sprint. Selecting validated records ensures consistency in features and targets while avoiding the reconstruction or imputation of unverifiable information. With a scope of 30 sprints, the evaluation focuses on prototype validation within the context of the company's internal data, allowing model results to be interpreted in alignment with the available data.

Table 1. Description of the research dataset

Unit of Analysis	Sprint Level
Data Coverage	30 sprint tervalidasi dari 23 proyek internal
Independent Variables (X)	`sprint_duration_days` and the quantity of specific team roles (after One-Hot Encoding)
Dependent Variable (Y)	WBS_Estimation`; also recorded as `Actual_Cost` with an identical value in the data export
Exclusion Criteria	WBS entries lacking complete sprint details, specifically duration and/or role composition

The pre-processing stage begins by verifying the completeness of the required values for each sprint. Textual data regarding role composition is then converted into numerical variables using One-Hot Encoding, allowing differences in role presence or quantity to serve as model features. Feature scaling is not applied because the three algorithms tested are tree-based or rely on combinations of tree-based predictions and do not require scaling to determine splits. The target variable remains in Rupiah so that the output can be directly interpreted as a WBS estimate. To provide context for generalization, the characteristics of the internal dataset are structurally compared against the public benchmarks listed in Table 2 [31], [32], [33].

Table 2. Structural comparison of internal dataset and public benchmark

Dataset	Granularitas	Size	Key Target/Attribute	Relevansi terhadap Studi
Company-internal	Sprint	30 sprint / 23 proyek	WBS cost; sprint duration; role composition	Most suitable for a WBS-based DSS within the company's internal context; the data is proprietary.
Albrecht	Project	24 proyek	Function points/size dan effort	Classic effort benchmark; units and targets are not identical to sprint WBS costs.
Maxwell	Project	62 proyek	Size, personnel, schedule/environmental factors, actual effort	Project-level benchmarks; does not provide the same sprint WBS structure
ISBSG D&E	Development and Enhancement Project	13.147 proyek D&E (rilis 2025)	Size, effort, schedule, platform, development techniques	Large external benchmark base; schema mapping and data access required prior to head-to-head testing.

The comparison in Table 2 is methodological rather than a comparison of performance figures. Albrecht and Maxwell focus on project-level effort, whereas ISBSG maintains a much larger repository but employs schemes and attributes that do not align perfectly with the internal WBS data [31], [32], [33]. As the current research data export does not provide equivalent feature mappings across these benchmarks, neither data merging nor head-to-head testing was performed to avoid an "apples-to-oranges" comparison; external validation following schema alignment has been designated as a topic for future research.

Model validation employed 3-fold cross-validation with randomization, meaning that in each iteration, approximately two-thirds of the data were used for training and one-third for testing. MAPE was selected as the primary criterion for model selection due to its provision of an easily interpretable relative error measure; RMSE served as the nominal error measure (in Rupiah), while R^2 was treated as a diagnostic metric for generalization capability rather than a measure of effect size. The Random Forest model was selected as the final model if it yielded the lowest MAPE, with RMSE and R^2 used to verify the consistency of this decision. Given the sample size (n=30) and the high number of encoded features, performance interpretation was contextualized within the internal dataset, with due regard for sensitivity to fold composition [25], [26], [27], [28], [29], [30]. To ensure reporting remained based on verifiable outputs, fold-level mean ± standard deviation statistics for MAPE and R^2 were not reconstructed, as the processed data export contained neither fold identifiers nor per-fold metric logs. As a readily available measure of dispersion, the study reports the mean ± standard deviation of the error per sprint for Random Forest predictions in the results section.

RESULT AND DISCUSSION

Comparative Analysis of Machine Learning Automation Against the Company's WBS Baseline

This study positions the organization's historical WBS_Estimation as the target for replicating expert judgment. In the exported dataset used, the Actual_Cost column contains values identical to the WBS_Estimation across all 30 sprints; consequently, an Error_WBS of 0% represents definitional consistency with the historical target rather than a predictive metric against independent realized costs. Given this setup, the Machine Learning model focuses on replicating the Project Manager's WBS estimates based on the available sprint data patterns. A comparison of WBS values and Random Forest predictions across all sprints is presented in Table 3.

Table 3. Comparison of WBS values and random forest predictions across 30 sprints

Sprint_ID	Actual_Cost	WBS_Estimation	ML_Prediction_RF	Error_WBS (%)	Error_ML_RF (%)
PROYEK_57_SPRINT_1	88821250	88821250	84872577.35	0.0	4.445640
PROYEK_57_SPRINT_2	101072500	101072500	82814115.90	0.0	18.064641
PROYEK_56_SPRINT_1	100975000	100975000	73231959.03	0.0	27.475158
PROYEK_55_SPRINT_1	69444735	69444735	71988401.84	0.0	3.662865

Sprint_ID	Actual_Cost	WBS_Estimation	ML_Prediction_RF	Error_WBS (%)	Error_ML_RF (%)
PROYEK_55_SPRINT_2	68235031	68235031	76279650.25	0.0	11.789574
PROYEK_55_SPRINT_3	95755869	95755869	83809886.93	0.0	12.475457
PROYEK_53_SPRINT_1	58821994	58821994	71323442.55	0.0	21.253017
PROYEK_52_SPRINT_1	76338500	76338500	83806907.55	0.0	9.783278
PROYEK_16_SPRINT_1	71297100	71297100	75165136.44	0.0	5.425237
PROYEK_13_SPRINT_1	61317018	61317018	63043496.70	0.0	2.815660
PROYEK_13_SPRINT_2	61317018	61317018	62593636.86	0.0	2.081998
PROYEK_51_SPRINT_1	76338500	76338500	78087785.60	0.0	2.291485
PROYEK_20_SPRINT_1	78447500	78447500	83700316.66	0.0	6.695964
PROYEK_17_SPRINT_1	87174500	87174500	96488903.13	0.0	10.684780
PROYEK_12_SPRINT_1	73565298	73565298	75213146.43	0.0	2.239981
PROYEK_12_SPRINT_2	73565298	73565298	76168839.38	0.0	3.539089
PROYEK_15_SPRINT_1	73565298	73565298	74765628.41	0.0	1.631653
PROYEK_54_SPRINT_1	81655215	81655215	73345798.73	0.0	10.176222
PROYEK_54_SPRINT_2	81655215	81655215	75025644.52	0.0	8.118980
PROYEK_19_SPRINT_1	132730000	132730000	86067222.94	0.0	35.156164
PROYEK_18_SPRINT_1	76589566	76589566	71463858.23	0.0	6.692436
PROYEK_18_SPRINT_2	73565298	73565298	72988844.32	0.0	0.783595
PROYEK_14_SPRINT_1	61317018	61317018	63062987.58	0.0	2.847447
PROYEK_4_SPRINT_1	55344094	55344094	88393862.84	0.0	59.716885
PROYEK_5_SPRINT_1	77080949	77080949	132306233.13	0.0	71.645828
PROYEK_3_SPRINT_1	46573647	46573647	104945471.63	0.0	125.332304
PROYEK_6_SPRINT_1	202096690	202096690	56104892.61	0.0	72.238589
PROYEK_45_SPRINT_1	49182078	49182078	99175924.34	0.0	101.650537
PROYEK_46_SPRINT_1	106491810	106491810	141526738.59	0.0	32.899177
PROYEK_41_SPRINT_1	82033249	82033249	55103477.87	0.0	32.827873

Based on the 30 rows in Table 3, the Random Forest percentage absolute deviation shows a mean of 23.55% (standard deviation: 31.69%), a median of 9.98%, and a range of 0.78%–125.33%. Most sprints exhibit relatively low error rates compared to a few extreme cases, while some sprints show deviations exceeding 100%. This distribution underscores the importance of interpreting performance at the sprint level and maintaining Project Manager validation when the DSS is applied to diverse project characteristics. PROYEK_13_SPRINT_1 and PROYEK_13_SPRINT_2 are recorded as distinct Sprint_IDs; although they share the same WBS target, they yield different ML_Prediction_RF and Error_ML_RF values, reflecting the model's response to the specific feature combinations present in each sprint.

The practical implication of these results is that the DSS can assist in generating initial estimates based on historical patterns in a structured, human-in-the-loop manner. Project Managers retain the role of validating business context, sprint characteristics, and high-deviation predictions, ensuring that the model's output serves as analytical support to strengthen the WBS estimation process using the company's internal data.

Machine Learning Algorithm Performance Evaluation

Three regression architectures Decision Tree Regressor, Random Forest Regressor, and Hybrid Model (Voting Regressor) were evaluated using 3-fold cross-validation. In accordance with the criteria established in the Methodology, MAPE was used as the primary metric for model selection, RMSE as the measure of nominal error, and R^2 as a diagnostic indicator of generalization capability. A summary of the metrics for the three models is presented in Table 4.

The results in Table 4 indicate that the Random Forest Regressor is the best-performing model among the three candidates: it achieved the lowest MAPE (23.548050%) and RMSE (Rp35,273,980), as well as the highest R^2 value (-0.535332—closest to zero) compared to the Decision Tree and Hybrid Model. Since all three criteria point to the same candidate, the Random Forest was selected as the final model for the study.

Table 4. Summary of global comparative metric evaluation

Estimation Method / Approach	MAPE (%)	RMSE (Rp)	R^2 Score
Decision Tree Regressor	32.335201	47.316.490	-1.762602
Random Forest Regressor	23.548050	35.273.980	-0.535332
Hybrid Model (Voting Regressor)	26.788135	39.866.070	-0.961102

The manual WBS baseline was not included as a predictive model in Table 4 because the target dataset was derived from the same WBS values; consequently, the baseline's MAPE of 0 and R^2 of 1 reflect the data construction rather than predictive capability regarding an independent target. Among the machine learning models, all R^2 values from the testing phase were negative. In this evaluation, a negative R^2 indicates that the model's squared error exceeds that of a simple predictor based on the target mean. Therefore, the Random Forest's R^2 of -0.535 serves as a diagnostic indicator that there is still room to improve generalization, even though the Random Forest model demonstrated the best performance among the three candidates based on MAPE, RMSE, and R^2.

The combination of a 23.55% MAPE, negative R^2, error standard deviation per sprint, and internal data coverage suggests that the Random Forest model is best suited as a DSS prototype within a human-in-the-loop framework. The findings support the viability of using Random Forest as a DSS prototype to learn and replicate internal WBS patterns within the scope of the research data, with Project Managers retaining the responsibility of validating the generated predictions. Comparisons with the Albrecht, Maxwell, and ISBSG models in this study focused on dataset characteristics (Table 2); numerical cross-dataset testing would require prior mapping of equivalent features and targets [31], [32], [33].

Analysis of Feature Contribution (Feature Importance) and Expert Logic

The research's theoretical contribution focuses on translating sprint-level Agile management parameters into a Work Breakdown Structure (WBS) cost estimation framework capable of non-linear learning. To identify the features contributing most significantly to the formation of Random Forest splits, Feature Importance values were extracted and are presented in Figure 2. This metric serves as an indicator of the relative importance of features within the model, rather than as a directional coefficient or evidence of a causal relationship.

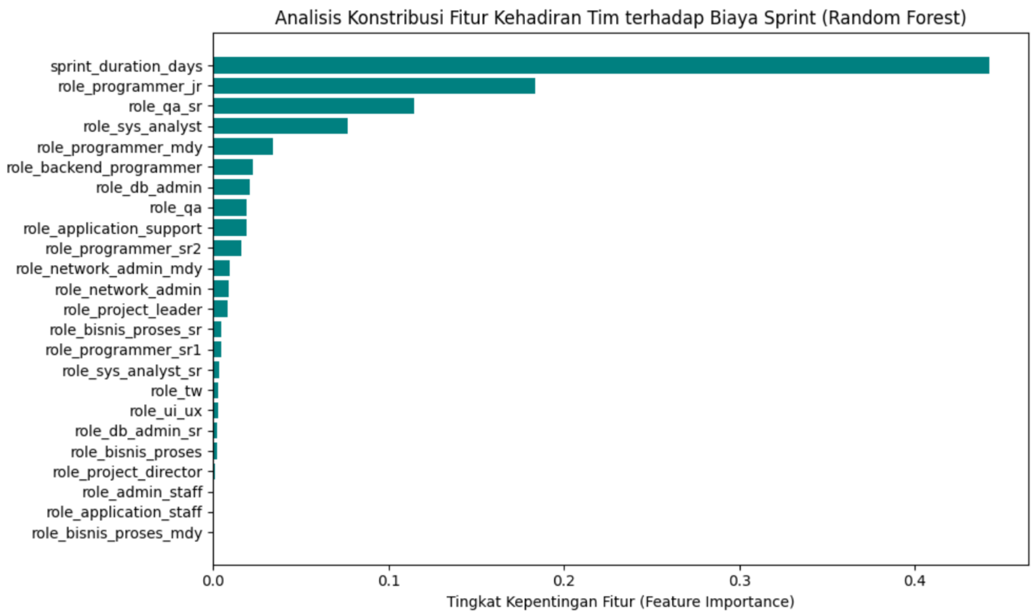

Figure 2. Analysis of team presence feature contribution to sprint cost (Random Forest)

The visualization in Figure 2 shows that `sprint_duration_days` is the most dominant feature, followed by `role_programmer_jr`, `role_qa_sr`, and `role_sys_analyst`. This ranking indicates that the split decisions in the Random Forest model are primarily influenced by variations in sprint duration, followed by variations in team role composition. Consequently, the interpretation of feature importance focuses on the relative significance of the features in shaping WBS cost predictions across the 30 analyzed sprints.

The findings can be analyzed through two mutually reinforcing lenses as follows:

1) Machine Learning Algorithmic Mechanics. Random Forest constructs multiple decision trees and captures patterns through non-linear splits and feature interactions. Features exhibiting variations that are more informative for reducing impurity are assigned higher importance. Consequently, the dominance of `sprint_duration_days` and specific role features reflects a combination of frequency,

variation, and interaction across the 30 sprints; this ranking is interpreted as a reflection of model behavior rather than a causal relationship [25], [26], [27], [28], [29], [30].

2) WBS Management Realities. Sprint duration and role composition are operational parameters that can jointly alter cost requirements for each sprint. In the model results, `role_programmer_jr` and `role_qa_sr` emerged as the role features with the highest importance after `sprint_duration_days`, followed by `role_sys_analyst`. This ranking indicates that WBS cost patterns are learned from the joint variation between duration and team composition; thus, cost estimation relies not merely on a single standard rate assumption but on non-linear patterns formed by the combination of available features.

Thus, the framework's novelty lies in operationalizing Agile parameters at the sprint level—specifically duration and role composition into a non-linear Work Breakdown Structure (WBS) cost mapping that is interpretable via Feature Importance. This interpretability provides Project Managers with insights into which features most strongly influence the model's decisions, while complementing analyses of pricing policies, work complexity, and expert validation regarding contextual factors beyond the model's features.

CONCLUSION

Based on the computational experiments conducted, the research objective was achieved through the development of a data-driven Decision Support System (DSS) designed to replicate sprint-level WBS estimates using 30 validated sprints from 23 internal projects. Among three candidates, the Random Forest Regressor was selected as the final model, yielding the lowest MAPE (23.548050%), the lowest RMSE (IDR 35,273,980), and the highest R^2 (-0.535332). The negative R^2 value and the error standard deviation per sprint of 31.69% led to the model being positioned as a decision support prototype within the internal data context, with potential for performance enhancement through data expansion and validation in future research.

Feature importance analysis identified `sprint_duration_days` as the dominant feature, followed by `role_programmer_jr` and `role_qa_sr`. These findings support the study's primary contribution: translating sprint duration and role composition into a non-linear WBS cost estimation structure at the sprint granularity level. Practically, the DSS functions as a preliminary decision support tool within a human-in-the-loop framework; the model provides estimates based on historical patterns, while the Project Manager reviews business context, sprint characteristics, and predictions showing high deviation.

Within the scope of this study, validation was performed on 30 validated sprints, as only these records contained sufficient detail regarding duration and role composition. `WBS_Estimation` served as the target variable; in the exported data, its values were identical to the `Actual_Cost` column, meaning the evaluation focused on the model's ability to replicate historical WBS estimates rather than on independent realized costs. As fold identifiers/logs were unavailable in the output, MAPE and R^2 statistics per fold could not be reconstructed; furthermore, head-to-head performance testing against public benchmarks was not conducted because the feature and target schemes were not equivalent to those of the internal dataset. Despite these scope limitations, the study successfully provides preliminary evidence of the internal DSS prototype's feasibility and establishes a clear direction for future development. Future research could expand the dataset by incorporating as many validated sprints as data availability allows, including independent realized costs where available, recording metrics for each fold and considering repeated or nested cross-validation, and performing external validation after mapping the schema to benchmarks such as Albrecht, Maxwell, or ISBSG.

REFERENCES

[1] J. Pasuksmit, P. Thongtanunam, and S. Karunasekera, “A Systematic Literature Review on Reasons and Approaches for Accurate Effort Estimations in Agile,” *ACM Comput. Surv.*, vol. 56, no. 11, pp. 1–37, Nov. 2024, doi: 10.1145/3663365.

[2] B. Alsaadi and K. Saeedi, “Ensemble effort estimation for novice agile teams,” *Inf. Softw. Technol.*, vol. 170, p. 107447, Jun. 2024, doi: 10.1016/j.infsof.2024.107447.

[3] M. Iqbal *et al.*, “Exploring issues of story-based effort estimation in Agile Software Development (ASD),” *Sci. Comput. Program.*, vol. 236, p. 103114, Sep. 2024, doi: 10.1016/j.scico.2024.103114.

[4] J. Cubillos, J. Aponte, D. Gomez, and E. Rojas, “Agile effort estimation in Colombia: An assessment and opportunities for improvement,” *Sci. Comput. Program.*, vol. 236, p. 103115, Sep. 2024, doi: 10.1016/j.scico.2024.103115.

[5] L. Fürst, T. Hovelja, M. Poženel, and D. Vavpotič, "Impact of competence on agile effort estimation in academic setting," *Softw. Pract. Exp.*, vol. 54, no. 11, pp. 2144–2159, Nov. 2024, doi: 10.1002/spe.3333.

[6] W. Rosa and S. Jardine, "A novel method to early agile effort estimation through functional initiatives," *Journal of Systems and Software*, vol. 223, p. 112302, May 2025, doi: 10.1016/j.jss.2024.112302.

[7] W. Rosa and S. Jardine, "Data-driven agile software cost estimation models for DHS and DoD," *Journal of Systems and Software*, vol. 203, p. 111739, Sep. 2023, doi: 10.1016/j.jss.2023.111739.

[8] O. R. Yürüm, H. Ünlü, and O. Demirörs, "An alternative software benchmarking dataset: effort estimation with machine learning," *Journal of Systems and Software*, vol. 231, p. 112591, Jan. 2026, doi: 10.1016/j.jss.2025.112591.

[9] M. Hassanali, M. Soltanaghaei, T. Javdani Gandomani, and F. Zamani Boroujeni, "Software development effort estimation using boosting algorithms and automatic tuning of hyperparameters with Optuna," *Journal of Software: Evolution and Process*, vol. 36, no. 9, Sep. 2024, doi: 10.1002/smr.2665.

[10] L. Cao, "Estimating Efforts for Various Activities in Agile Software Development: An Empirical Study," *IEEE Access*, vol. 10, pp. 83311–83321, 2022, doi: 10.1109/ACCESS.2022.3196923.

[11] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: A systematic performance evaluation," *Softw. Pract. Exp.*, vol. 52, no. 1, pp. 39–65, Jan. 2022, doi: 10.1002/spe.3009.

[12] N. Gupta and R. P. Mahapatra, "Automated software effort estimation for agile development system by heuristically improved hybrid learning," *Concurr. Comput.*, vol. 34, no. 25, Nov. 2022, doi: 10.1002/cpe.7267.

[13] B. Yalçıner, K. Dinçer, A. G. Karaçor, and M. Ö. Efe, "Enhancing Agile Story Point Estimation: Integrating Deep Learning, Machine Learning, and Natural Language Processing with SBERT and Gradient Boosted Trees," *Applied Sciences*, vol. 14, no. 16, p. 7305, Aug. 2024, doi: 10.3390/app14167305.

[14] M. Turic, S. Celar, S. Dragicevic, and L. Vickovic, "Advanced Bayesian Network for Task Effort Estimation in Agile Software Development," *Applied Sciences*, vol. 13, no. 16, p. 9465, Aug. 2023, doi: 10.3390/app13169465.

[15] E. Rodríguez Sánchez, E. F. Vázquez Santacruz, and H. Cervantes Maceda, "Effort and Cost Estimation Using Decision Tree Techniques and Story Points in Agile Software Development," *Mathematics*, vol. 11, no. 6, p. 1477, Mar. 2023, doi: 10.3390/math11061477.

[16] N. Govil and A. Sharma, "Estimation of cost and development effort in Scrum-based software projects considering dimensional success factors," *Advances in Engineering Software*, vol. 172, p. 103209, Oct. 2022, doi: 10.1016/j.advengsoft.2022.103209.

[17] S. A. Butt *et al.*, "Prediction based cost estimation technique in agile development," *Advances in Engineering Software*, vol. 175, p. 103329, Jan. 2023, doi: 10.1016/j.advengsoft.2022.103329.

[18] S. A. Butt *et al.*, "A software-based cost estimation technique in scrum using a developer's expertise," *Advances in Engineering Software*, vol. 171, p. 103159, Sep. 2022, doi: 10.1016/j.advengsoft.2022.103159.

[19] Meiliana, G. Daniella, N. Wijaya, N. G. E. Putra, and R. Efata, "Agile Software Development Effort Estimation based on Product Backlog Items," *Procedia Comput. Sci.*, vol. 227, pp. 186–193, 2023, doi: 10.1016/j.procs.2023.10.516.

[20] M. Fernandez-Diego, E. R. Mendez, F. Gonzalez-Ladron-De-Guevara, S. Abrahao, and E. Insfran, "An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 166768–166800, 2020, doi: 10.1109/ACCESS.2020.3021664.

[21] C. Eduardo Carbonera, K. Farias, and V. Bischoff, "Software development effort estimation: a systematic mapping study," *IET Software*, vol. 14, no. 4, pp. 328–344, Aug. 2020, doi: 10.1049/iet-sen.2018.5334.

[22] A. Jadhav and S. K. Shandilya, "Reliable machine learning models for estimating effective software development efforts: A comparative analysis," *Journal of Engineering Research*, vol. 11, no. 4, pp. 362–376, Dec. 2023, doi: 10.1016/j.jer.2023.100150.

[23] S. P. Shankar, S. S. Chaudhari, V. Mishra, and T. Stephan, "Intelligent techniques for predictive analytics in Agile software development," *Sci. Rep.*, vol. 16, no. 1, p. 11195, Feb. 2026, doi: 10.1038/s41598-026-41102-4.

[24] W. Long, Z. Qixin, and Y. Luxia, “Software effort estimation based on inception network optimized by enhanced banyan tree growth optimizer,” *Sci. Rep.*, vol. 16, no. 1, p. 143, Dec. 2025, doi: 10.1038/s41598-025-29028-9.

[25] S. Cateni, V. Colla, and M. Vannucci, “Improving the Stability of the Variable Selection with Small Datasets in Classification and Regression Tasks,” *Neural Process. Lett.*, vol. 55, no. 5, pp. 5331–5356, Oct. 2023, doi: 10.1007/s11063-022-10916-4.

[26] C. H. Rashid *et al.*, “Software Cost and Effort Estimation: Current Approaches and Future Trends,” *IEEE Access*, vol. 11, pp. 99268–99288, 2023, doi: 10.1109/ACCESS.2023.3312716.

[27] M. Rahman, H. Sarwar, MD. A. Kader, T. Gonçalves, and T. T. Tin, “Review and Empirical Analysis of Machine Learning-Based Software Effort Estimation,” *IEEE Access*, vol. 12, pp. 85661–85680, 2024, doi: 10.1109/ACCESS.2024.3404879.

[28] T. R. Benala, A. Kaushik, S. Dehuri, and L. C. Jain, “Computational intelligence for estimating software development effort: a systematic mapping study,” *Iran Journal of Computer Science*, vol. 7, no. 3, pp. 607–630, Sep. 2024, doi: 10.1007/s42044-024-00178-9.

[29] E. I. Mustafa and R. Osman, “A random forest model for early-stage software effort estimation for the SEERA dataset,” *Inf. Softw. Technol.*, vol. 169, p. 107413, May 2024, doi: 10.1016/j.infsof.2024.107413.

[30] R. Celkevicius and R. F. S. M. Russo, “An integrated model for allocation and leveling of human resources in IT projects,” *International Journal of Managing Projects in Business*, vol. 11, no. 2, pp. 234–256, May 2018, doi: 10.1108/IJMPB-09-2016-0074.

[31] A. J. Albrecht and J. E. Gaffney, “Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation,” *IEEE Transactions on Software Engineering*, vol. SE-9, no. 6, pp. 639–648, Nov. 1983, doi: 10.1109/TSE.1983.235271.

[32] Katrina. D. Maxwell, *Applied Statistics for Software Managers*. Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[33] International Software Benchmarking Standards Group, “Development and Enhancement Data,” 2025. Accessed: Aug. 18, 2026. [Online]. Available: https://www.isbsg.org/development-and-enhancement-data/