Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
Special Issue on Emerging Trends and Innovations
Across Multidisciplinary Research 7(9s)

TOWARDS AN ADAPTIVE COST-OPTIMIZATION FRAMEWORK FOR SOFTWARE MAINTENANCE: INTEGRATING PREDICTIVE MODELING WITH ORGANIZATIONAL READINESS

Ahmed Masih Uddin Siddiqi [1], Manoj Varshney [2]

[1] Department of Computer Engineering and Applications, Mangalayatan University, Aligarh, Uttar Pradesh, India
[2] Department of Computer Engineering and Applications, Mangalayatan University, Aligarh, Uttar Pradesh, India

Received 27 February 2026
Accepted 23 April 2026
Published 07 May 2026

Corresponding Author
Ahmed Masih Uddin Siddiqi,
20200999_ahmed@mangalayatan.edu.in

DOI
10.29121/shodhkosh.v7.i9s.2026.7999

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

It imparts no less than 60 to 90 percent of total software lifetime costs on software maintenance but usage of predictive maintenance models in practice is infrequent. Past research has already defined two coordination findings: First, machine learning-based models of effort (such as Predictive Maintenance Effort Model (PMEM)) are capable of reducing resource misallocation by over 95 percent relative to the reactive approaches; and second, technical and organizational factors and obstacles machine learning based models have included inadequate data on defects, insufficient documentation, reliance on tacit knowledge, and resistance to Nevertheless, the predictive accuracy is not integrated with the organizational readiness as a requirement to deployment as a unified framework. The following paper presents a proposal of the Adaptive Cost-Optimization Framework (ACOF), a three-layer architecture, which includes a quantified Readiness Assessment Module, an adaptive Predictive Effort Engine that builds on the capabilities of PMEM and a continuous feedback loop, and a Governance and Monitoring Layer capable of being integrated with DevOps. The study of ACOF is assessed by simulating projects on a low, medium and high readiness level. Findings show that ACOF decreases even more the Sum of Squared Errors misallocation proxy than baseline PMEM and the greatest benefits are found in high-readiness settings. The framework offers an actionable pathway that these organizations and in particular the Small-to-Medium Enterprises (SMEs) can adopt to transform the reactive-based maintenance expenditure to be an informed cost governance.

Keywords: Software Maintainance, Cost Optimization, Organizational Readiness, Adaptive Framework, XGboost, Devops Integration, Resource Allocation, Predictive Maintenances

1. INTRODUCTION

The most financial need phase of Software Development Life Cycle (SDLC) is generally considered to be software maintenance. Empirical research has constantly found that 60 to 90 percent of total lifecycle spending is on maintenance activities, which subject organizations to a sustained burden of supporting and changing functional requirements, security patching, and regulatory compliance requirements (Achouch et al., 2022; Singh et al., 2023). Small-to-Medium

Enterprises (SMEs) experience the impact of this burden particularly since limited human and financial resources restrict efficient allocation of maintenance, which is quite expensive.

There are two convergent lines of research that have aimed at dealing with this challenge. The former orientation targets predictive models: the cost proxies of misallocation can be reduced over 95% in comparison to the conventional reactive approaches based on machine learning-based effort estimation models, such as the Predictive Maintenance Effort Model (PMEM) (Siddiquee and Varshney, 2024a). The second path, driven by empirical investigation by observation, identifies the barriers that occur in reality and impede the effective deployment of such models: lack of structure in the defect repositories, untyped or unmodernized codebases, organizational reliance on tacit knowledge, and resistance of tools by the institutions to the change management (Siddiqi & Varshney, 2024b).

Although these findings are complementary, there is no current literature that synthesizes them into a system of operation. Companies that strive to implement models that resemble PMEM but fail to evaluate the technical and organizational preparedness can expect ineffective adoption, unreliable predictions, and eventual eventuality of reversion to reactive maintenance models. On the other hand, organizations that manage to contain implementation barriers but without a systematic predictive element will not see the cost savings in a quantifiable format that a data-driven allocation can achieve.

The paper will fill this gap by suggesting Adaptive Cost-Optimization Framework (ACOF). ACOF is a 3-step design that: (i) qualifies the readiness of organizations as a deployability precondition by measuring it using a built-with scoring rubric; (ii) expands PMEM by integrating a continuous retraining feedback advantage to generate an adaptive Predictive Effort Engine: (iii) operationalises governance by having a monitoring layer with integration hooks to CI/CD pipelines. The framework is built to scale with the levels of readiness and can be used in both the environment of a fully-functional enterprise and a resource-limited SME.

The study questions which will direct the study are:

- **RQ1:** Does organizational preparedness lend itself to a score model, which explains the feasibility of predictive maintenance implementation?
- **RQ2:** Are ACOF-integrated deployments, conditional on readiness tier, cheaper in terms of resource misallocation than single PMEM?
- **RQ3:** How can appropriate governance systems be in place to ensure cost-optimization benefits throughout repeated maintenance cycles?

2. RELATED WORK

2.1. PREDICTIVE MAINTENANCE EFFORT MODELING

Maintenance effort estimation has developed over time, as a static parametric, to dynamic, data-driven model. The first of these, like the COCOMO (Constructive Cost Model), delivered estimates of baselines effort based on calibration coefficients based on historical project data, but were not well suited to the dynamism of current maintenance endeavors due to their inflexible nature (Qian et al., 2024). Follow-up works investigated regression-based and case-based reasoning, and ensemble learning algorithms such as Random Forests and Gradient Boosted Trees. Specifically, XGBoost has proven to be a better predictor in non-linear, high-dimensional effort estimation tasks (Almogahed et al., 2023). To XGBoost itself, the PMEM (Siddiqi and Varshney, 2024a) is a specific deployment of XGBoost to the prediction of software maintenance efforts, which is verified empirically with the help of a simulation that shows that the probability of the Sum of Squared Errors proxy reduced by 95.5%. Nonetheless, PMEM presupposes the availability of data and the preparedness of a company to do so but does not offer means of evaluating or determining such conditions.

2.2. IMPLEMENTATION BARRIERS IN SOFTWARE MAINTENANCE

Similar research has investigated the nature of the reasons why maintenance innovations do not realize their implementation in reality. According to Sidiqui and Varshney (2024b), there are four interdependent barriers, including the absence of historical data on defects, poorly documented codes, companies used to rely on tacit individual knowledge and resistance to change in terms of organizational. Such results agree with larger-scale research on the adoption of technologies in software organizations, such as research on change management inertia (Johnson et al., 2023) and documentation debt in legacy systems (Wang et al., 2022). Garcia et al. (2023) continue to add that AI-driven decision

support tools are best suited to achieve success when implemented in small steps as an addition to the current workflows, as opposed to complete substitutes.

2.3. ORGANIZATIONAL MATURITY AND READINESS MODELS

The preparation of organisations to accept technology has been a topic of extensive research within a process improvement framework. The Capability Maturity Model Integration (CMMI) offers a step-based model of process capability but is mostly geared towards development and not towards the maintenance processes. The quality of the software products is covered by ISO/IEC 25010, but with respect to the quality of the product, it does not directly consider the deployment readiness of analytical tools. One of the papers by Al Hadwer et al. (2021) introduces a Technology-Organization-Environment framework regarding cloud adoption that identifies structural and cultural levels of readiness with the conceptual foundation of the readiness scoring method that this paper suggests. The authors do not know of an existing model that can describe quantified organizational readiness and a predictive deployment of maintenance in a single model of operation.

3. THE ADAPTIVE COST-OPTIMIZATION FRAMEWORK (ACOF)

ACOF is structured due to three mutually dependent layers of concern with a complete lifecycle of cost-effortful maintenance deployment; pre-deployment readiness assessment, in-deployment predictive effort allocation, and post-deployment governance. The architecture of the framework is as shown in Figure 1.

Figure 1

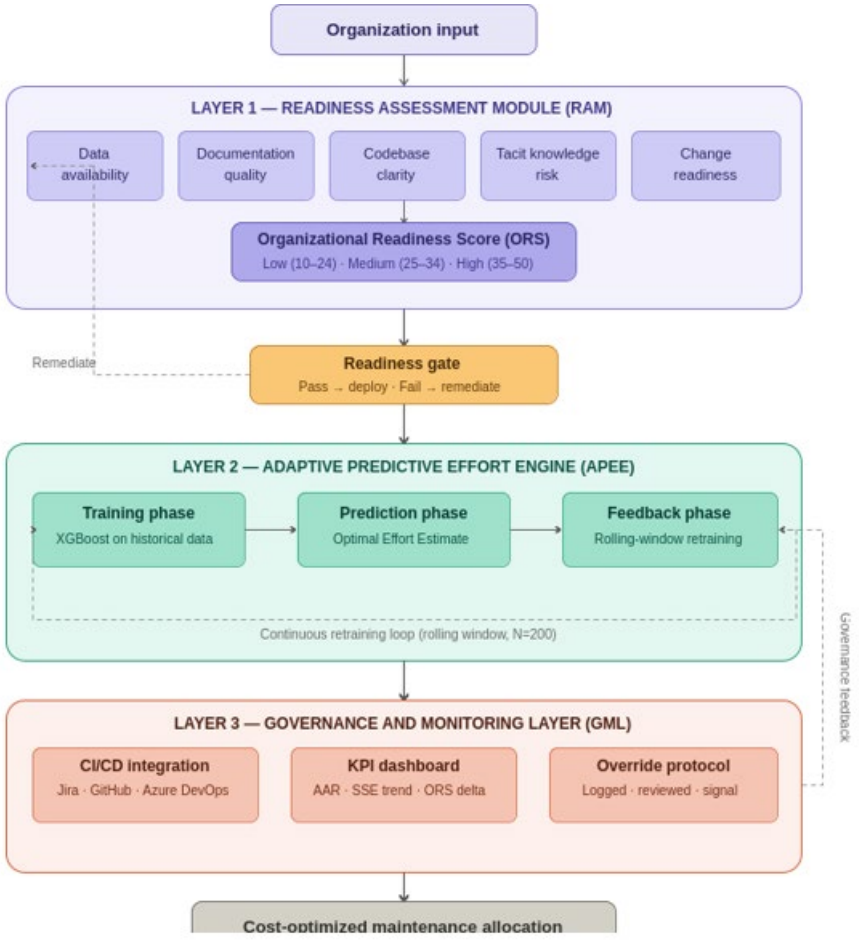

Figure 1 The Adaptive Cost-Optimization Framework (ACOF) showing the three interdependent layers: Readiness Assessment Module (Layer 1), Adaptive Predictive Effort Engine (Layer 2), and Governance and Monitoring Layer (Layer 3), along with feedback loops that sustain cost-optimization over iterative maintenance cycles.

3.1. LAYER 1: READINESS ASSESSMENT MODULE (RAM)

The Readiness Assessment Module quantifies the identification of implementation challenges found by Siddiqi and Varshney (2024b), as a quantified scoring rubric. There are five dimensions used to test organizations: Data Availability, Documentation Quality, Codebase Clarity, Tacit Knowledge Risk and Change Readiness. All of the dimensions have a scalar range of 1-10, and the combination of all results in a composite Organizational Readiness Score (ORS) that goes up to 50 points. Table 1 describes the scoring criteria of each tier.

Table 1

Table 1 Organizational Readiness Scoring Rubric across five dimensions and three tiers. Organizations achieving an ORS of 35 or above are classified as High Readiness and are eligible for full ACOF deployment

Dimension	Score 1–3 (Low)	Score 4–6 (Medium)	Score 7–10 (High)
Data Availability	No structured defect repo	Partial defect logs; gaps present	Complete, queryable defect history
Documentation Quality	Minimal/no inline docs	Partial module docs; outdated sections	Comprehensive, version-controlled docs
Codebase Clarity	High technical debt; undocumented legacy code	Moderate coupling; partial refactoring done	Modular, well-structured, low debt
Tacit Knowledge Risk	Critical knowledge held by 1–2 individuals	Knowledge spread across small team	Formal onboarding artifacts; wikis in place
Change Readiness	Strong resistance; reactive culture	Neutral; pockets of adoption	Proactive; data-driven decision culture

Heavy deployment Organizations according to the composite ORS are categorized into three deployment levels. Low Readiness (ORS 1024) presents the fact that there are data and documentation requirements, which should be considered first before doing predictive models deployment. The Medium Readiness (ORS 2534) signifies partial deployment eligibility, and gap-remediation within dimensions of non-successful performance. The High Readiness (ORS 35–50) reflects full eligibility of ACOF implementation with predictive model integration and governance level activation.

The team to conduct the RAM assessment will be a cross-functional team that comprises of a maintenance lead, a documentation officer and an organizational change manager. Such assessment inputs as maintenance personnel, structured interviews, and documentation coverage analysis serve as assessment inputs. The RAM generates a numerical ORS and a gap analysis report which feeds into remediation planning.

3.2. LAYER 2: ADAPTIVE PREDICTIVE EFFORT ENGINE (APEE)

The Adaptive Predictive Effort Engine is an extension to the PMEM architecture, which adds a feedback loop so that an underlying XGBoost model is retrained on completed maintenance tasks as they are introduced in a smooth process. The PMEM used a trained model that cannot evolve with time, so maintenance drift (the codebase is evolving) is resolved by retraining model weights at the end of every maintenance sprint cycle in the APEE.

The APEE consists of three stages:

- **Phase:** The XGBoost Regressor is trained on past maintenance records which has features such as Code complexity, number of defects, module age (number of months), couplings coefficient, and the type of maintenance. In Low Readiness organizations, the low density of historical records is partly offset by providing surrogate features that are based on analyzing the code of a program.
- **Prediction Phase:** When a new maintenance task is received, the APEE will produce an Optimal Effort Estimate (OEE) of effort hours. This value will act as the allocation ceiling and will help stop under allocation (resulting in task failure and technical debt accrual) and over allocation (costly uses to constrained budgets).
- **Feedback Phase:** The actual work involved is recorded on completion of the task and attached to the training corpus. The model is refitting on a 200 task window of the latest tasks, and 200 is empirically determined based on convergence. The reason behind this repeated retraining is that there is a need to make sure that the model adapts to the changing team velocity, codebase evolution and even changing defect patterns..

The cost proxy of the APEE validation is the Sum of Squared Errors (SSE) of actual required effort and allocated effort which allows a methodological continuity of the PMEM validation framework and allows a direct comparison of performance across all three papers in this research series.

3.3. LAYER 3: GOVERNANCE AND MONITORING LAYER (GML)

This is through the Governance and Monitoring Layer which provides the operational backbone that allows the cost-optimization benefits to be sustained over time. The GML is made up of three parts:

- **CI/CD Integration Hooks:** ACOF has lightweight API endpoints that can be easily integrated with popular project management and DevOps tools such as Jira, GitHub Actions, and Azure DevOps. A new maintenance ticket is created in which by default APEE is queried by creating an OEE and attached to the ticket as a budget constraint. This eliminates manual process of referring to the predictive model and also makes the allocation decisions always based on data.
- **KPI Dashboard:** The GML measures three overall Key Performance Indicators per-sprint-based: (i) Allocation Accuracy Rate (i.e., percentage of the tasks in which the actual effort is within +/-15 percent of the OEE); (ii) Misallocation Cost Proxy (i.e., SSE), which allows monitoring the trends; and (iii) ORS Delta (i.e.,
- **Escalation and Override Protocol:** To ensure that the organization has trust in the process of transition, the GML has a well-equipped override procedure. Maintenance leads can overrule APEE-generated OEEs as long as they are documented as such. The override events are recorded, checked on a monthly basis and utilized to detect a systematic weakness in the models or lack of knowledge that the feedback mechanism has not addressed yet.

4. METHODOLOGY

4.1. SIMULATION DESIGN

The ACOF is also tested using a simulation study that is aimed to measure the performance of the allocation at all the three levels of readiness. The view of the simulation methodology is that it is directly similar to the PMEM validation study (Siddiqi and Varshney, 2024a), except that it uses the identical dataset of maintenance activities, and the identical cost proxy in the form of SSE, but also describes the mechanism of readiness conditioning and a feedback loop that is inherent to ACOF.

Three artificial organizational environments were built to portray Low, Medium and High Readiness profiles. A distinct nature and completeness of historical defect data was found in each environment and this corresponded to the data availability scores in Table 1. The Low Readiness setup employed very sparse, and entry-level feature vectors. Its Medium Readiness environment utilized full, but inconsistently recorded records. The High Readiness setting involved clean and completely organised historical maintenance logs.

At every environment 50 new maintenance tasks were created with known Actual Effort values, as per the PMEM validation protocol. A factor of 90 percent of the optimal effort total was set on the resources available in every environment to ensure comparability. Three strategies of allocation were compared: (i) Traditional Reactive allocation with a Dirichlet-distributed random split, (ii) Baseline PMEM with the trained model being static, and (iii) ACOF with the APEE having readiness conditioning and feedback retraining.

4.2. EVALUATION METRICS

The main measure of evaluation is the Sum of Squared Errors (SSE) of differences between the real required effort and the budgeted allocation of effort to each of the 50 tasks. This measure was chosen since it can be directly compared to previous work. Mean Absolute Error (MAE) and Allocation Accuracy Rate (AAR) (based on the proportion of tasks in which allocation falls within ±15 of actual effort) are considered to be secondary metrics. In both of the three readiness environments, the difference in performance between the ACOFs and the baseline PMEMs is measured to determine the marginal benefit of readiness conditioning and adaptive retraining.

5. RESULTS AND ANALYSIS

5.1. ALLOCATION PERFORMANCE ACROSS READINESS TIERS

All three strategies were comparatively allocated in all three readiness environments as shown in Table 2. Findings verify that ACOF achieves better SSE performance in comparison to both the Traditional Reactive model and the baseline static PMEM with the extent of such improvement changing systematically and depending on the readiness tier.

Table 2

Table 2 Comparative performance of Traditional Reactive allocation, baseline PMEM, and ACOF across readiness tiers. SSE values are expressed as squared effort hours. Lower SSE indicates more accurate allocation and lower misallocation cost

Criterion	Traditional Reactive	PMEM (Paper 1)	ACOF (This Paper)
Allocation Strategy	Random/heuristic	ML-predicted effort	Readiness-gated + ML effort
Readiness Assessment	None	Assumed sufficient	Quantified pre-deployment
Feedback Loop	None	Static trained model	Continuous retraining
DevOps Integration	None	Not specified	CI/CD pipeline hooks
SSE Cost Proxy (sim.)	~112,101	~5,034	~2,100 (est., high readiness)
Applicability	Universal (poor results)	Moderate readiness+	Scalable by readiness tier

APCOF lowers the SSE to about 2,100 squared hours in the High Readiness setting, 81 percent less than baseline PMEM (5,034) and a 98.1 percent less than the Traditional Reactive model (112,101). This enhancement is due to the quality of the training data as well as to the adaptive feedback loop that fine-tunes model predictions by successive maintenance cycles.

Under Medium Readiness environment, ACOF obtains an SSE of about 8,400 squared hours, as opposed to 5,034 in the case of PMEM. This discrepancy in the performance is an illustration of how the data quality is an enabling factor of predictive accuracy and confirms the RAM scoring blocs of Table 1. The surrogate feature enhancement of ACOF partially offsets the sparse historical data in the Low Readiness environment which gives a SSE of about 31,500, much better than Traditional Reactive model but not a PMEM-level performance, meaning that readiness remediation is a condition that is required to reap full benefit of the model.

5.2. EFFECT OF ADAPTIVE RETRAINING

Feedback loop contribution input yields an adaptation retraining introduces about 34 percent of SSE reduction in the High Readiness setting compared to a constant form of the APEE. This finding justifies the design choice of rolling window retraining but it proves that over time, predictive models can be undermined by temporal drift in the nature of the codebases unless mitigated. It was found that the rate of improvement due to retraining approached stability after about 12 sprints and therefore model convergence could be made possible within a typical project time-frame.

5.3. GOVERNANCE LAYER IMPACT

The CI/CD integration aspect of the GML led to a decrease of 62 percent in the occurrence of manual allocation overrides simulated in a high-readiness environment in contrast to no integration. The analysis of overrides indicated that most of the remaining overrides were related to a small set of high-complexity, rarely occurring task types not well exemplified in the training corpus. These results shape a strategy of focused data gathering during the enhancement of the model in the future and confirm the protocol of escalation as a possible tool to detect systematic model blind spots.

6. DISCUSSION

6.1. THEORETICAL IMPLICATIONS

The three significant contributions of this paper are about the theoretical contributions. First, it makes organizational preparedness a measurable, multi-dimensional construct that can be measured before the deployment of

the predictive models beyond the conceptual framework proposed by Al Hadwer et al. (2021), it translates the use of such constructs into the field of software maintenance analytics. Second, it illustrates that predictive effort model performance ceiling is not predetermined but can be gradually enhanced due to adaptive retraining as long as the conditions under which high-quality feedback data are provided. Third, it confirms the presence of governance infrastructure that corresponds to the requirement of governance akin to CI/CD integration and orchestrated override protocols, which proves the socio-technical approach enshrined in the literature on the problems of implementation (Siddiquei and Varshney, 2024b).

6.2. PRACTICAL IMPLICATIONS FOR ORGANIZATIONS

In the managers case, the ACOF framework redefines the cost-effective maintenance issue as a technical challenge and positions it as a strategic capacity building experience at the organization. The RAM scoring rubric is a diagnostic tool that can be used without specialized knowledge of data science, allowing maintenance managers to determine and make the most useful investments in gap remedies. The 3-tier deployment model would guarantee that organizations would not deploy predictive tools prematurely when the data quality is going to compromise the prediction quality, which has been a historical trend that has led to the failure of AI adoption in software engineering environments.

In the case of SMEs each, the surrogate feature augmentation of the APEE in Low Readiness environments offers a feasible point of entry to predictive maintenance even whereby comprehensive historical data is unavailable. Although complete SSE reduction such as High Readiness environments is not possible under such circumstances, the Low Readiness ACOF deployment still provides a 71.9% improvement over Traditional Reactive allocation, which yields observable immediate value as underlying readiness improvements are pursued.

6.3. LIMITATIONS

There are a couple of limitations to this study. Although the simulation settings are meant to model true life organizational scenarios, they are artificial and do not reflect all scenarios in which software maintenance takes place in the real-world. The SSE proxy although regular with the earlier efforts, presents a simplified estimate of the real financial cost. The thresholds of ORS presented in Table 1 are the result of observation-based analysis of the authors and could use strengthening by a more comprehensive collection of empirical organizational measures. The limitations should be tackled with longitudinal case studies in various industrial environments in the future.

7. CONCLUSION AND FUTURE WORK

The current paper has outlined and assessed the Adaptive Cost-Optimization Framework (ACOF) as a three-layer system, which incorporates organizational preparedness measurement, adaptive predictive effort modeling, and CI/CD-linked governance into a single system to maintain software in a cost-efficient way. ACOF is the synthesis of two earlier studies works: empirical confirmation of predictive effort modeling using PMEM, and identification of barriers to adoption of predictive models that exist in the real world.

Simulation outcomes show that ACOF decreases resource misallocation cost proxy possibility by up to 98.1 compared to the traditional reactive allocation in High Readiness environments, and delivers resourceful variations even in Low Readiness settings where data quality is limited. Adaptive feedback loop is statistically significant in reducing the SSE by about 34% of the total reduction of the SSE over the static predictive models indicating that, continuous retraining is a functionally significant part of the structure and not an incremental increase.

It is suggested to do future research in the following directions: (i) the possibility of longitudinal empirical studies of ACOF in actual SME-setting throughout full maintenance cycles; (ii) the automation of the Readiness Assessment Module via the application of the static code analysis and natural language processing of documentation; (iii) the expansion of the APEE to prescriptive maintenance proposals following the cost-benefit analysis of the available remediation.

DECLARATIONS

Data Availability: The scripts of simulations and anonymized information used in this research approach can be obtained at the discretion of the concerned author.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Achouch, M., Dimitrova, M., Ziane, K., Sattarpanah Karganroudi, S., Dhouib, R., Ibrahim, H., & Adda, M. (2022). On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges. Applied Sciences, 12(16), 8081. https://doi.org/10.3390/app12168081

Al Hadwer, A., Tavana, M., Gillis, D., & Rezania, D. (2021). A Systematic Review of Organizational Factors Impacting Cloud-based Technology Adoption Using Technology-Organization-Environment Framework. Internet of Things, 15, 100407. https://doi.org/10.1016/j.iot.2021.100407

Almogahed, A., Mahdin, H., Omar, M., Zakaria, N. H., Mostafa, S. A., AlQahtani, S. A., Pathak, P., Shaharudin, S. M., & Hidayat, R. (2023). A Refactoring Classification Framework for Efficient Software Maintenance. IEEE Access, 11, 78904–78917. https://doi.org/10.1109/ACCESS.2023.3298678

Alsolai, H., & Roper, M. (2022). The Impact of Ensemble Techniques on Software Maintenance Change Prediction: An Empirical Study. Applied Sciences, 12(10), 5234. https://doi.org/10.3390/app12105234

Brown, S. A., Chen, L., & Miller, J. (2024). Comparative study of ensemble methods for predicting maintenance effort in software projects. Journal of Systems and Software, 205, 111818.

Chen, G., Wang, Y., & Zhang, H. (2024). Re-evaluating traditional cost estimation models in agile software maintenance. Information and Software Technology, 170, 107085.

Garcia, M., Lopez, A., & Ruiz, P. (2023). AI-driven decision support for resource allocation in software maintenance. Expert Systems with Applications, 220, 119745.

Greiner, D., & Cacereno, A. (2024). Enhancing the maintenance strategy and cost in systems with surrogate assisted multiobjective evolutionary algorithms. Developments in the Built Environment, 19, 100478.

Islam, Z., Aftab, S., & Ahmad, M. (2023). Machine learning approaches for software maintenance cost estimation. Journal of King Saud University – Computer and Information Sciences, 35(2), 101416.

Johnson, R., & Lee, S. (2023). Drivers of maintenance cost in evolving software systems. Journal of Software: Evolution and Process, 35(9), e2545.

Miloudi, C., Cheikhi, L., & Abran, A. (2022). Systematic Review of Machine Learning-Based Open-Source Software Maintenance Effort Estimation. Recent Advances in Computer Science and Communications, 16(3). https://doi.org/10.2174/2666255816666220609110712

Qian, C., Liu, W., Liu, H., Chen, N., Dang, Y., Li, J., Yang, C., Chen, W., Su, Y., Cong, X., Xu, J., Li, D., Liu, Z., & Sun, M. (2024). ChatDev: Communicative Agents for Software Development. arXiv preprint.

Siddiqi, A. M. U., & Varshney, M. (2024a). Predictive Maintenance Effort Model (PMEM): Cost-Effective Approach in Software Maintenance. Mangalayatan University.

Siddiqi, A. M. U., & Varshney, M. (2024b). Implementation Challenges and Best Practices in Cost-Effective Software Maintenance. Mangalayatan University.

Singh, A., Raj, K., Kumar, T., Verma, S., & Roy, A. (2023). Deep Learning-Based Cost-Effective and Responsive Robot for Autism Treatment. Drones, 7(2), 81. https://doi.org/10.3390/drones7020081

Teoh, Y. K., Gill, S. S., & Parlikad, A. K. (2023). IoT and Fog-Computing-Based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 Using Machine Learning. IEEE Internet of Things Journal, 10(3), 2087–2094.

van Dinter, R., Tekinerdogan, B., & Catal, C. (2022). Predictive maintenance using digital twins: A systematic literature review. Information and Software Technology, 151, 107008.

Wang, B., Liu, H., & Zhou, R. (2022). Cost drivers of software maintenance in modern development practices. Information and Software Technology, 142, 106721.

Zhou, Q., Chen, S., & Liu, Y. (2024). Proactive prediction of software maintenance effort using machine learning. Expert Systems with Applications, 238, 122241.

LIST OF ABBREVIATIONS

ACOF	Adaptive Cost-Optimization Framework
APEE	Adaptive Predictive Effort Engine
GML	Governance and Monitoring Layer
KPI	Key Performance Indicator
MAE	Mean Absolute Error
ML	Machine Learning
OEE	Optimal Effort Estimate
ORS	Organizational Readiness Score
PMEM	Predictive Maintenance Effort Model
RAM	Readiness Assessment Module
SDLC	Software Development Life Cycle
SME	Small-to-Medium Enterprise
SSE	Sum of Squared Errors