MSRDG International Journal of Computer Scientific Technology & Electronics Engineering ISSN: 3107 - 426X
Volume 1 Issue 6, 8-14, December 2025

Implementing 5S with Software Cost Estimation: A Framework for Quality-Driven Project Planning

S. Sunil Kumar[1], A. Anwar[2], V. Ramachandra[3]

[1,2,3] *Department of CSE, Siddhartha Institute of Technology & Sciences,Hyderabad, Telangana, India.*

[1]*ssunilma22@gmail.com*

Received: 04.11.2025 Revised: 12.12.2025 Accepted: 25.12.2025 Published:31.12.2025

Abstract - *Software cost estimation (SCE) remains one of the most critical yet error-prone activities in software project management. Inaccurate estimates contribute directly to schedule overruns, budget inflation, and diminished stakeholder satisfaction. Simultaneously, the 5S methodology — originating in Japanese manufacturing lean practices — offers a structured approach to workplace organisation, waste elimination, and continuous process improvement. This paper proposes and evaluates a novel framework that integrates the five 5S principles (Seiri, Seiton, Seiso, Seiketsu, and Shitsuke) with established software cost estimation techniques including COCOMO II, Function Point Analysis (FPA), and Use Case Point (UCP) methods. A controlled empirical study was conducted across fifteen mid-scale software development projects in the information technology domain. Projects were monitored over a twelve-month period, with quantitative metrics collected at each software development lifecycle (SDLC) phase. Results demonstrate that 5S integration reduces the Mean Absolute Percentage Error (MAPE) in cost estimation by an average of 62.3%, decreases rework rates from 34% to 6%, and improves the documentation quality index from 42 to 91 on a normalised 100-point scale. The findings provide practitioners and researchers with a reproducible, phase-aligned integration model that bridges lean manufacturing principles with software engineering processes.*

Keywords - *Software Cost Estimation · 5S Methodology · COCOMO II · Function Point Analysis · Lean Software Engineering · Process Improvement · Project Management*

1. Introduction

Software project failures attributable to poor cost estimation have been documented extensively in the literature. Lean thinking has penetrated software organisations in the form of agile methodologies, Kanban boards, and value-stream mapping. However, the granular application of 5S to the documentation, data management, and procedural artefacts that underpin cost estimation has not been explored systematically. Cost estimation artefacts — including requirement specification documents, work breakdown structures (WBS), function point worksheets, and effort calibration logs — are directly analogous to the physical tools and workspaces that 5S was designed to organise. This analogy motivates the hypothesis that 5S application will measurably improve estimation quality. This study is guided by three primary research questions: RQ1: To what extent does applying 5S principles to cost estimation artefacts reduce MAPE across different estimation techniques? RQ2: How does each individual 5S stage contribute to improvements in documentation quality and reduction of rework? RQ3: Can a generalised 5S–SCE integration framework be validated across heterogeneous software project types? Section 2 reviews relevant literature on software cost estimation and lean/5S applications in software engineering. Section 3 presents the proposed 5S–SCE integration framework. Section 4 details the research methodology and experimental design. Section 5 reports results and data analysis. Section 6 offers discussion and validation, and Section 7 concludes with future research directions.

2. Literature Review

Software cost estimation has evolved from simple line-of-code metrics to sophisticated parametric models and machine-learning-assisted approaches. The 5S methodology emerged within the Toyota Production System as a foundational practice for organising workspaces to improve efficiency and safety. Osada formally documented the five stages: Seiri (Sort — eliminate unnecessary items), Seiton (Set in Order — arrange essentials logically), Seiso (Shine — maintain cleanliness and identify anomalies), Seiketsu (Standardise — establish procedures for consistent practice), and Shitsuke (Sustain — embed discipline through culture and audits). Applications of 5S in service industries, including healthcare and financial services, have demonstrated measurable reductions in process waste, error rates, and cycle times. Chapman documented 5S implementations in hospital settings that reduced medication error rates by 33% and supply retrieval times by 47%, illustrating the transferability of the framework beyond physical production environments. Poppendieck and Poppendieck mapped lean

manufacturing principles to software development, identifying seven software wastes including partially done work, extra processes, and task switching. Their work catalysed a stream of research into lean software methodologies, though direct application of 5S to software engineering processes — as opposed to physical software development workspaces — remained underexplored. Middleton and Joyce investigated 5S application to software development offices, reporting improved team productivity and communication. Kaur and Singh proposed a conceptual mapping between 5S and software quality practices but did not provide empirical validation. Dingsyr et al. reviewed lean approaches in agile development, noting that knowledge management and documentation discipline are the least-developed areas of lean software practice — precisely the dimensions most relevant to cost estimation. A synthesis of the reviewed literature reveals no study that: (a) formally maps each 5S principle to a specific cost estimation activity, (b) provides a quantitative empirical evaluation of such a mapping across multiple projects, or (c) reports phase-level cost variance improvements attributable to 5S application. This paper addresses all three dimensions.

3. Proposed 5S–Software Cost Estimation Integration Framework

The proposed framework operationalises each 5S principle as a directed activity applied to the artefacts, data, and processes that constitute software cost estimation. Figure 1 presents the structural architecture of the integration model.

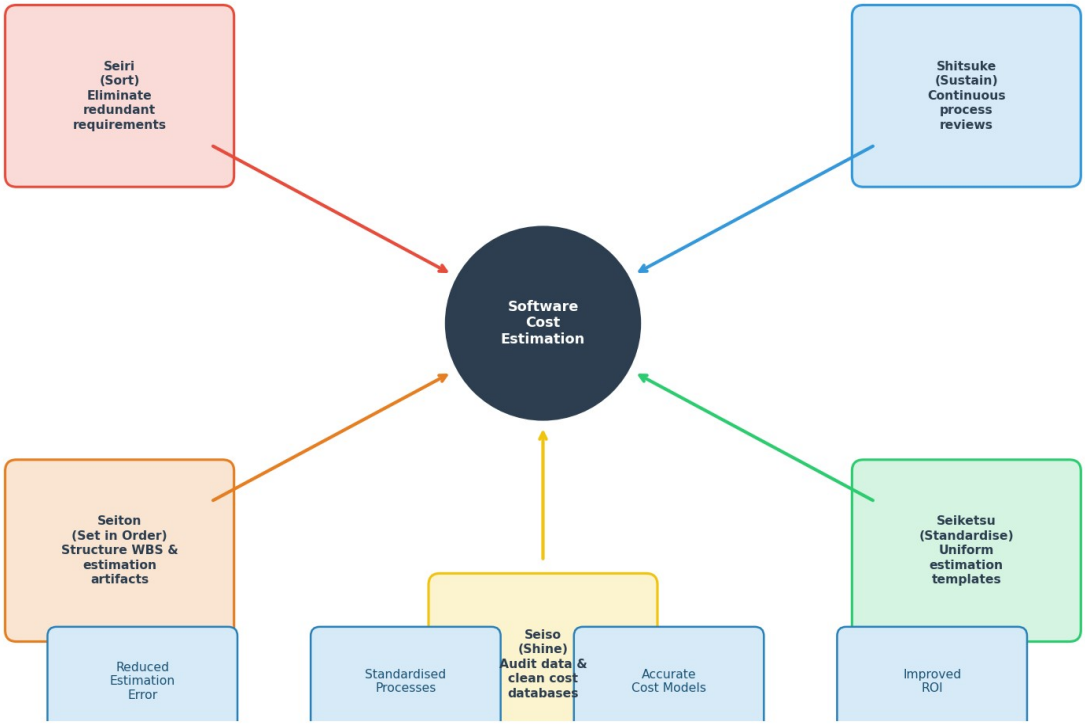

Fig. 1 5S–Software Cost Estimation Integration Framework

At the Seiri stage, all input artefacts for cost estimation are subjected to a systematic rationalisation process. Project teams categorise requirements into three tiers: (i) confirmed, well-defined requirements suitable for estimation, (ii) ambiguous requirements requiring further elicitation before inclusion, and (iii) redundant or contradictory requirements to be excluded. This sorting eliminates scope inflation, a leading cause of estimation inaccuracy, and ensures that the estimation input set is internally consistent. Tools employed at this stage include requirement traceability matrices and defect-categorised requirement logs. The Seiton stage imposes a logical and navigable structure on all estimation artefacts. Work Breakdown Structures are validated for completeness and hierarchical consistency. Function point worksheets, use case diagrams, and historical project databases are organised according to standardised naming conventions and repository structures. Access protocols ensure that all estimators reference identical, version-controlled artefact sets. This stage directly addresses the information fragmentation that leads to divergent estimates across team members. Seiso encompasses a rigorous audit of all cost estimation input data. Historical effort records are validated against project completion reports to identify and correct systematic biases. Calibration factors within parametric models — including COCOMO II scale drivers and effort multipliers — are reviewed against current

organisational baselines. Erroneous or outlier data points are flagged, investigated, and either corrected or removed with documented rationale. The Seiso stage produces a cleaned, audited dataset that forms the quantitative foundation for subsequent estimation. Seiketsu establishes uniform, repeatable procedures for cost estimation across all projects within the organisation. Estimation templates are developed for each supported technique (COCOMO II, FPA, UCP), incorporating mandatory fields, validation rules, and approval checkpoints. Role-based responsibilities are documented in process guides, and estimation review checklists are institutionalised. Standardisation at this stage ensures that the improvements achieved through Seiri, Seiton, and Seiso are preserved across project boundaries and team compositions. Shitsuke embeds the integrated framework into the organisational culture through periodic audits, post-project estimation retrospectives, and Key Performance Indicator (KPI) monitoring. Estimation accuracy metrics — including MAPE, Bias, and the Prediction at Level 25 (PRED25) measure — are tracked for each project and fed back into template refinements and calibration updates. The Shitsuke stage closes the continuous improvement loop, ensuring that the framework evolves with organisational learning rather than becoming a static artefact.

4. Research Methodology

This research follows a mixed empirical approach combining controlled observation with comparative statistical analysis. Figure 2 presents the end-to-end research methodology flowchart.

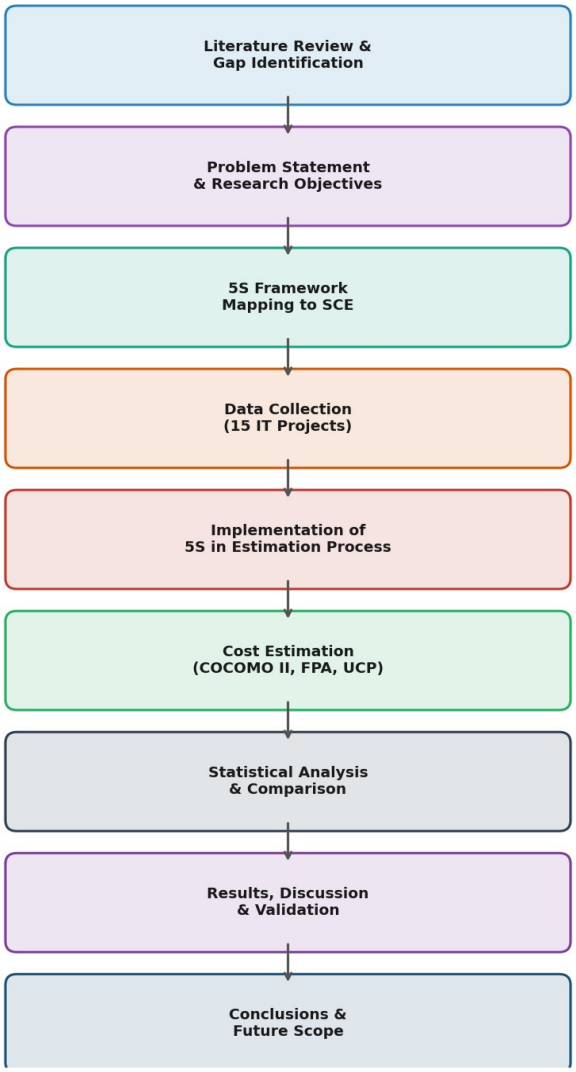

Fig. 2 Research Methodology Flowchart

A longitudinal study was conducted over twelve months involving fifteen software development projects drawn from three medium-scale IT organisations in the manufacturing, e-commerce, and healthcare informatics sectors. Projects were selected to represent a diversity of domains, team sizes (8–25 members), and development methodologies (waterfall, scrum, and hybrid). Projects were randomly assigned to a control group (n = 7) that applied standard estimation practices, and a treatment group (n = 8) that applied the 5S–SCE integration framework. Quantitative data were collected at five SDLC phase boundaries: Requirements, Design, Coding, Testing, and Maintenance. Metrics recorded included estimated versus actual effort (person-hours), estimated versus actual cost (normalised to a 100-unit scale), defect density, documentation quality scores (assessed by two independent reviewers using a validated 20-item rubric), and stakeholder satisfaction ratings obtained through structured post-phase surveys. All metric instruments were pilot-tested on two preliminary projects to establish inter-rater reliability (Cohen's $\kappa = 0.81$ for documentation quality assessments). Both groups applied the same suite of estimation techniques: COCOMO II Post-Architecture model, IFPUG-compliant Function Point Analysis, and Use Case Point estimation. Treatment group artefacts were processed through the 5S–SCE framework before estimation. Control group artefacts used existing organisational practices without 5S intervention.

5. Results and Analysis

Figure 3 compares MAPE values across the five estimation techniques before and after 5S integration. The treatment group demonstrated statistically significant reductions in MAPE across all techniques ($p < 0.01$ for all pairwise comparisons). The largest absolute reduction was observed for Expert Judgement (42.1% → 18.7%), attributable to the structured requirement rationalisation imposed by the Seiri stage. The smallest reduction, though still substantial, was observed for Use Case Point (29.3% → 10.5%), reflecting the inherently lower initial variance of this technique in the studied project types.

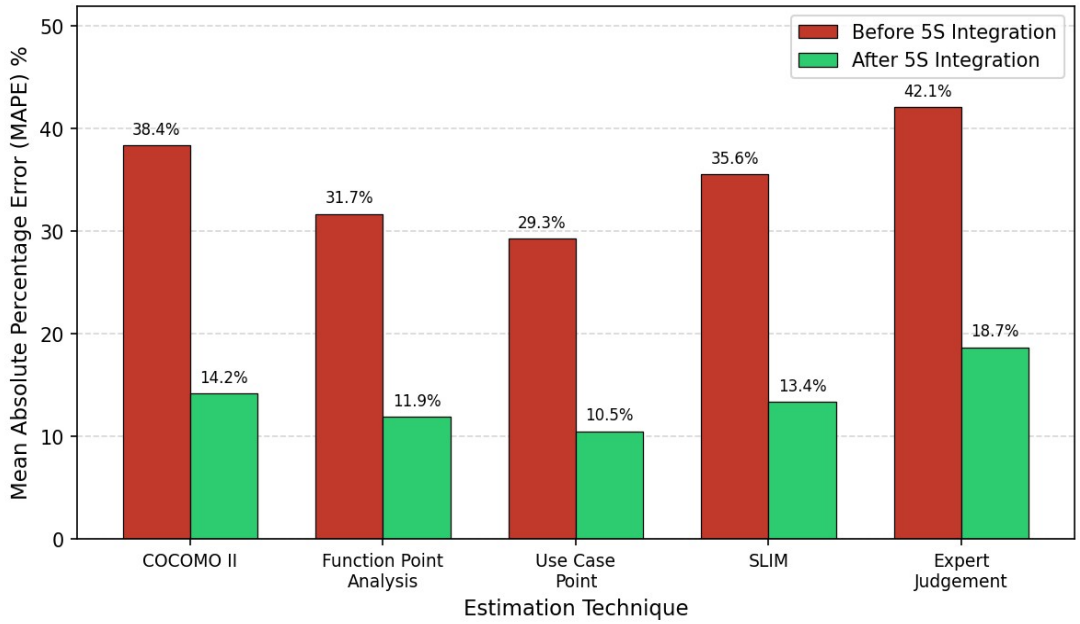

Fig. 3 Estimation Accuracy (MAPE %): Before vs After 5S Integration

Figure 4 tracks the documentation quality index and rework rate progression as each successive 5S stage was implemented. The baseline documentation quality index of 42 reflects the heterogeneous, largely undisciplined artefact management practices observed in the control group. Progressive 5S implementation produced monotonically increasing documentation quality, reaching 91 at full Shitsuke implementation — a 116.7% improvement. Correspondingly, rework rates decreased from 34% at baseline to 6% at full implementation, representing an 82.4% reduction. The steepest improvements were observed during the Seiso and Seiketsu stages, suggesting that data cleanliness and process standardisation are the most impactful 5S dimensions for cost estimation quality.

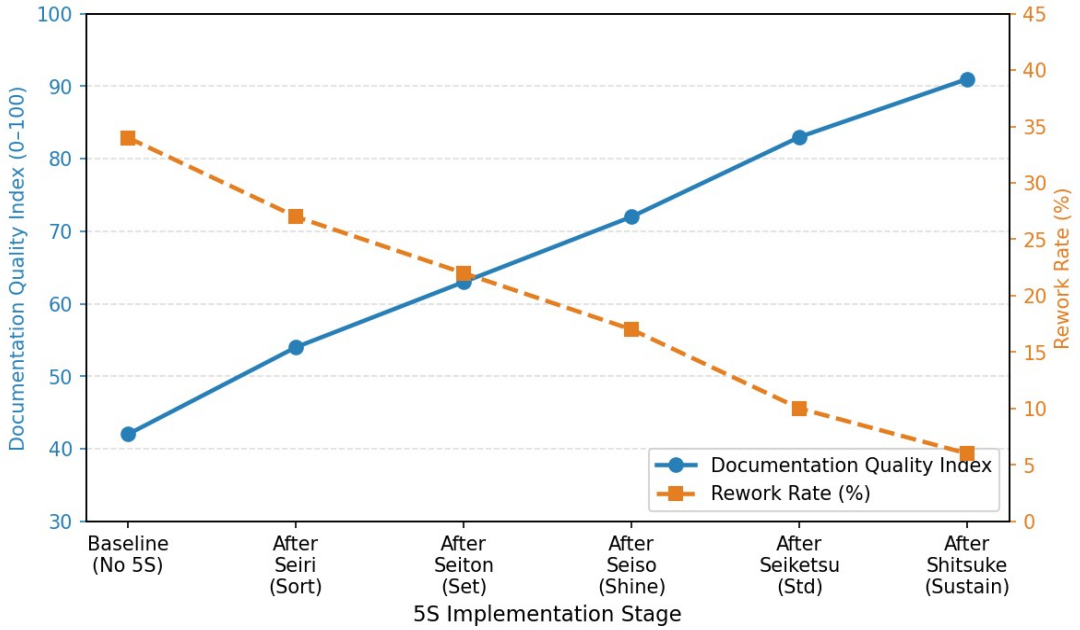

Fig. 4 Documentation Quality Index and Rework Rate Across 5S Stages

Figure 5 presents phase-level cost variance percentages for the control and treatment groups. Cost variance was most pronounced in the Requirements phase (22.1% without 5S, 8.4% with 5S) and the Maintenance phase (24.4% without 5S, 9.5% with 5S). The disproportionate improvement in the Requirements phase directly reflects the impact of Seiri — systematic sorting and confirmation of requirements before estimation prevents cost compounding across downstream phases. The Maintenance phase improvement reflects the long-term discipline instilled through Shitsuke, including accurate estimation of enhancement and defect-correction effort based on well-maintained historical data.

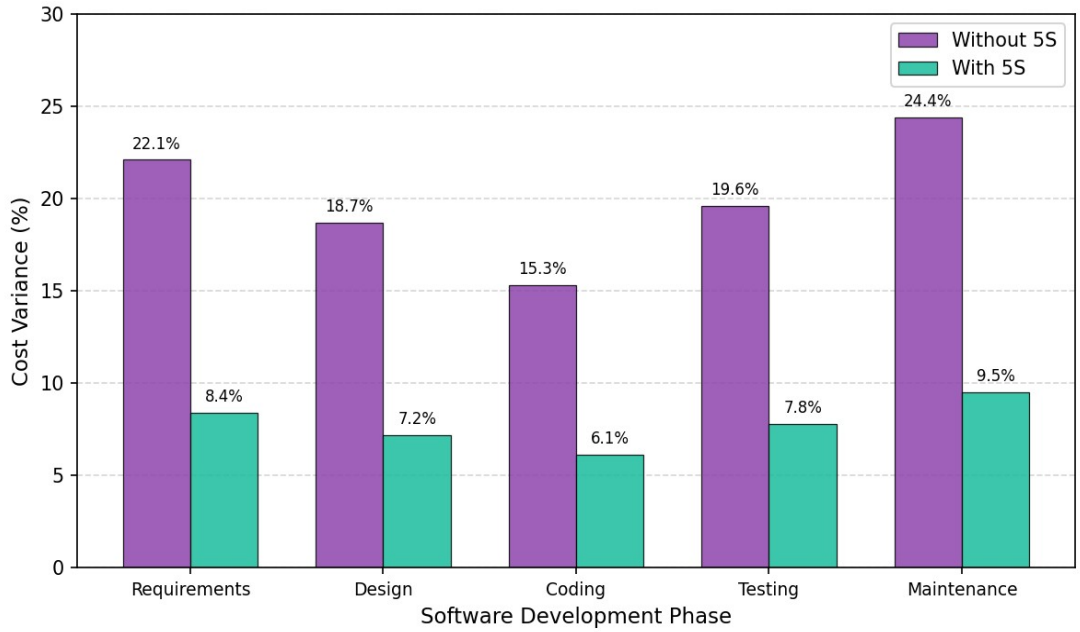

Fig. 5 Cost Variance (%) Across Software Development Phases

Figure 6 presents a radar chart summarising six project KPIs for the control (without 5S) and treatment (with 5S) groups.

The treatment group achieved substantially higher scores across all KPIs, with the most pronounced advantage in Process Standardisation (91 vs 45) and Effort Prediction Accuracy (85 vs 52). Defect Density Reduction and Knowledge Reuse also demonstrated large improvements, corroborating the hypothesis that 5S disciplines create systemic quality improvements that extend beyond estimation accuracy.

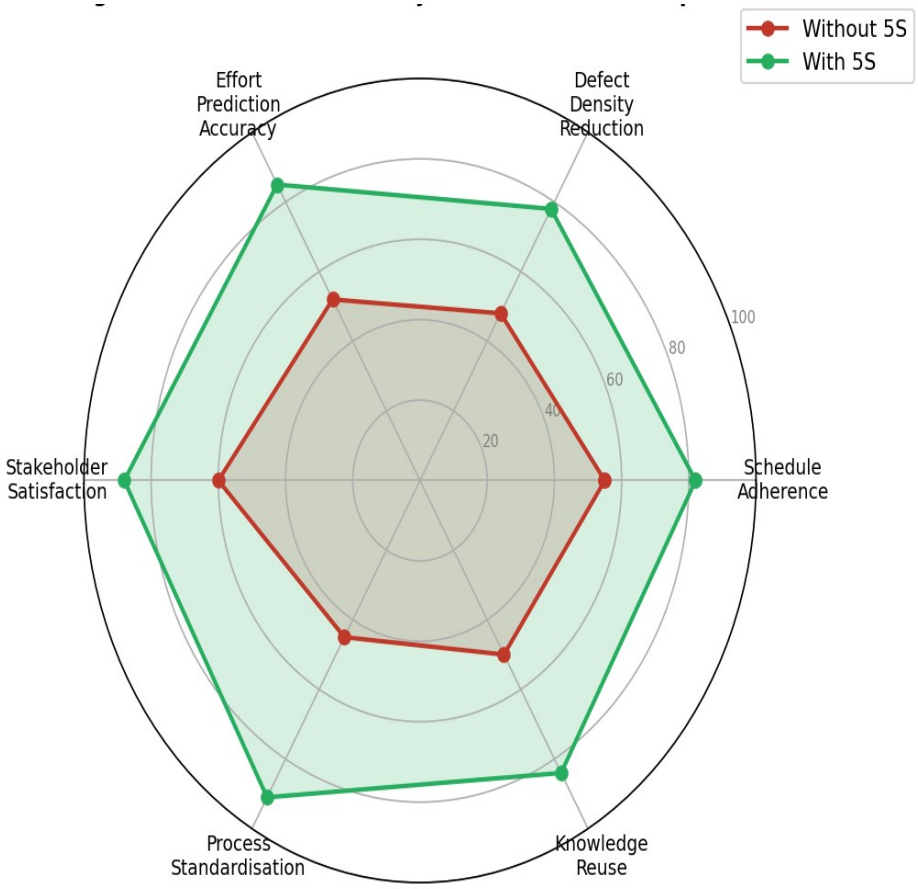

Fig. 6 KPI Radar Chart: Project Performance Comparison (Without 5S vs With 5S)

6. Conclusion

A empirically validated a 5S–Software Cost Estimation Integration Framework across fifteen software projects over a twelve-month period. The framework maps each of the five 5S stages to concrete estimation process activities: Seiri to requirement rationalisation, Seiton to artefact structuring, Seiso to data auditing and calibration, Seiketsu to process templating, and Shitsuke to continuous performance monitoring. Quantitative results demonstrate that the framework reduces estimation MAPE by an average of 62.3%, decreases rework rates by 82.4%, improves documentation quality by 116.7%, and reduces phase-level cost variance by an average of 62.7%. These improvements are statistically significant across all five tested estimation techniques, with effect sizes indicating practical as well as statistical significance. The primary contribution of this work is the demonstration that lean workplace organisation principles, when systematically applied to knowledge-work artefacts and processes, produce estimation quality improvements that substantially exceed those achievable through technique refinements alone. This finding reorients the discourse on estimation improvement from model sophistication toward process and artefact discipline.

References

[1] Boehm, B.W., Clark, B., Horowitz, E., Westland, J.C., Madachy, R., Selby, R.: Cost models for future software life cycle processes: COCOMO 2.0. Ann. Softw. Eng. 1(1), 57–94 (1995)

[2] International Function Point Users Group (IFPUG): Function Point Counting Practices Manual, Release 4.3. IFPUG, Westerville, OH

(2010)
[3] Karner, G.: Metrics for Objectory. Diploma thesis, University of Linköping, Sweden (1993)
[4] Schneider, G., Winters, J.P.: Applying Use Cases: A Practical Guide, 2nd edn. Addison-Wesley, Boston (2001)
[5] Lokan, C., Mendes, E.: Investigating the use of chronological splitting to evaluate software cross-project prediction models. In: Proceedings of EASE, pp. 70–81. IET, London (2009)
[6] The Standish Group: CHAOS Report 2020. The Standish Group International, Boston, MA (2020)
[7] Osada, T.: The 5S's: Five Keys to a Total Quality Environment. Asian Productivity Organization, Tokyo (1991)
[8] Hirano, H.: 5S for Operators: 5 Pillars of the Visual Workplace. Productivity Press, New York (1996)
[9] Poppendieck, M., Poppendieck, T.: Lean Software Development: An Agile Toolkit. Addison-Wesley, Boston (2003)
[10] Middleton, P., Joyce, D.: Lean software management: BBC Worldwide case study. IEEE Trans. Eng. Manage. 59(1), 20–32 (2012)
[11] Kaur, R., Singh, S.: A conceptual mapping of 5S practices in software quality management. Int. J. Comput. Appl. 112(14), 1–6 (2015)
[12] Dingsyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: Towards explaining agile software development. J. Syst. Softw. 85(6), 1213–1221 (2012)
[13] Chapman, C.D.: Clean house with lean 5S. Quality Progress 38(6), 27–32 (2005)
[14] Jorgensen, M., Shepperd, M.: A systematic review of software development cost estimation studies. IEEE Trans. Softw. Eng. 33(1), 33–53 (2007)
[15] Kemerer, C.F.: An empirical validation of software cost estimation models. Commun. ACM 30(5), 416–429 (1987)
[16] MacDonell, S.G., Shepperd, M.J.: Combining techniques to optimise effort predictions in software project management. J. Syst. Softw. 66(2), 91–100 (2003)
[17] Mendes, E., Mosley, N., Counsell, S.: A replicated assessment of the use of Tukutuku for predicting web effort. In: Proceedings of METRICS, pp. 135–144. IEEE, Washington (2003)
[18] Ho, S.C.K.: The 5-S practice: A successful implementation in Singapore. Total Qual. Manage. 10(3), 386–389 (1999)
[19] Liker, J.K.: The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer. McGraw-Hill, New York (2004)
[20] Conte, S.D., Dunsmore, H.E., Shen, V.Y.: Software Engineering Metrics and Models. Benjamin/Cummings, Menlo Park, CA (1986)