Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

RESEARCH ARTICLE

Data Integrity Validation in ERP and Middleware Modernization Programs

Sai Deekshith Katkam
Independent Researcher, USA
ORCID ID: 0009-0008-9123-7818
Corresponding Author: Sai Deekshith Katkam, **E-mail:** katkamsaideekshith@gmail.com

ABSTRACT

ERP and middleware modernization programs expose data integrity risks that conventional post-migration validation approaches are structurally unable to detect before production impact. These programs transform data schemas, middleware integration contracts, and business-process logic across interdependent system layers simultaneously, generating validation requirements that point-in-time audit methodologies cannot address at the required coverage depth. The TRACE-DI framework is proposed as a lifecycle-distributed validation protocol comprising five operational phases, Trace, Reconcile, Assert, Control, and Evidence, applied across six mandatory checkpoints from schema mapping completion through cutover readiness certification. Evaluation in enterprise ERP and middleware modernization contexts demonstrates improvements in data integrity accuracy across migration boundaries, reductions in reconciliation cycle time, and advances in audit evidence completeness. The framework provides modernization programmes with a structured, evidence-grounded validation approach compatible with phased migration strategies and parallel-run deployment models.

KEYWORDS

data integrity validation; ERP modernization; middleware transformation; TRACE-DI framework; migration testing; audit evidence

ARTICLE INFORMATION

ACCEPTED: 15 August 2026 **PUBLISHED:** 26 September 2026 **DOI:** 10.32996/jcsts.2026.8.9.16

1. Introduction

Among the most data-intensive undertakings in enterprise software, ERP modernization programs span platform migrations, cloud transformations, module replacements, and middleware re-platforming. Data integrity across these transformation boundaries determines the technical correctness of the modernized environment and, with equal importance, the accuracy of financial reporting, the reliability of operational decision-making, and the defensibility of audit trails required for regulatory compliance (Kulkarni & Bansal, 2023; Wanas et al., 2025). The costly remediation cycle, reconciliation backlog, and regulatory risk exposures which modernization efforts attempt to avoid have been the result of insufficiently rigorous data integrity validation prior to cutover (Seemanapalli, 2025).

There is always a difference between the validation approaches that are used for ERP modernization projects and the risks of integrity issues that come with these projects. The testing methods like extract compare verification, row count analysis, and sampling approach will work well in determining data loss at the gross level but will not be able to detect transformation problems, precision losses, referential integrity errors and audit log gaps resulting from migration software (Shivampeta, 2025; Bonthu, 2025).These missed defect categories carry a disproportionate share of post-cutover remediation cost precisely because they embed silently in data structures that normal system operation does not surface, persisting until a downstream process or audit procedure forces them into view.

Developed to address this gap, the TRACE-DI framework is a validation protocol comprising five operational phases applied across six mandatory checkpoints distributed through the modernization lifecycle. Trace establishes data lineage from source schema to target schema. Reconcile executes cross-system balance and volume comparison. Assert does business rule and referential integrity validation at the record level. Control ensures formal sign-off gates at each check point.Evidence generates audit-ready documentation packages throughout the lifecycle. The six checkpoints are positioned at schema mapping completion, data extraction validation, transformation rule verification, target load validation, parallel-run reconciliation, and cutover readiness certification.

Sections proceed as follows. Section 2 surveys relevant literature on data integrity validation, ERP migration risk, and middleware transformation challenges. Section 3 describes the TRACE-DI framework. Section 4 presents results in terms of integrity accuracy, reconciliation cycle time, and audit evidence completeness. Section 5 discusses implications and limitations. Section 6 presents the conclusion.

2. Literature Review

2.1 Data Integrity in Enterprise Information Systems

Data integrity is multidimensional in nature within the lifecycle of the record, which encompasses accuracy, consistency, completeness, timely nature, and validity of the records (Wanas et al., 2025). Data integrity is ensured in stable and non-changing environments by using constraints on the database, validating rules within the application, and reconciliation processes periodically. Modernization scenarios affect all these three ways of ensuring data integrity through simultaneous transformation (Seemanapalli, 2025; Shivampeta, 2025).

2.2 ERP Data Migration Risk and Validation Gaps

Data quality issues caused by violations of referential integrity resulting from an incomplete remapping of foreign keys during a schema migration comprise a large percentage of data quality problems after cutover (Kulkarni & Bansal, 2023). Precision and format errors in numeric columns because of type conversions in the ETL process lead to data records that validate against row counts but have erroneous data in the financial calculation columns (Bonthu, 2025). Auditing problems occur when migration software circumvents transaction processing and writes data records into the database without triggering audit logs (Wanas et al., 2025).Each of these failure modes reflects a distinct defect category requiring a distinct validation response.

2.3 Middleware Modernization and Semantic Integrity

Middleware modernization adds an integrity risk layer beyond the schema and record-level concerns of platform migration. When monolithic integration layers are replaced by service-oriented or event-driven middleware architectures, transformation logic previously embedded in integration adapters must be re-implemented in new service contracts and message mapping definitions (Bonthu, 2025; Seemanapalli, 2025). Errors in this re-implementation produce data corruption that is structurally invisible to record-level validation: records are present, field values are populated, referential constraints are satisfied, yet the semantic content of the data is incorrect because the transformation applied to it is wrong (Shivampeta, 2025). Detecting this class of error requires end-to-end business-process tracing from source event to target record, precisely the function served by the TRACE-DI Trace phase.

2.4 Audit Evidence Requirements in Modernization Programs

Tightening regulatory expectations around migration documentation, parallel-run evidence, and cutover certification artefacts have followed high-profile remediation cases in which organisations could not demonstrate the completeness and accuracy of post-migration data to auditors (KPMG, 2024; Deloitte, 2024). The TRACE-DI Evidence phase is designed to satisfy these requirements by generating structured, audit-ready documentation at each checkpoint rather than relying on retrospective reconstruction of validation activities.

2.5 Automated Reconciliation Tooling

Automated reconciliation tooling has advanced substantially over the prior decade, enabling large-scale record-by-record comparison across heterogeneous data stores at speeds impractical with manual sampling (Wanas et al., 2025; Seemanapalli, 2025). The TRACE-DI Reconcile phase leverages these capabilities while adding structured assertion logic and lineage tracing that pure reconciliation tooling does not provide, addressing volume and balance discrepancies through reconciliation, record-level logical errors through assertion, and transformation-logic errors through tracing.

3. Methodology

Five operational phases and six mandatory checkpoints organise the TRACE-DI framework. The phases define what validation activity is performed; the checkpoints define when in the modernization lifecycle each activity is mandatory. Every category of integrity risk is addressed at the point where it is most efficiently detected and remediated.

Table 1. TRACE-DI Framework: Five Phases and Primary Validation Activities

Phase	Full Name	Primary Activity	Key Artefact
Trace	Data Lineage Verification	Map every source field to its target counterpart; document transformation rule and derivation logic	Lineage map document
Reconcile	Cross-System Balance and Volume Comparison	Record-level, aggregate, and balance reconciliation between source and target	Reconciliation summary report
Assert	Business-Rule and Referential-Integrity Validation	Apply mandatory field, range, enumeration, and foreign-key constraints to all migrated records	Assertion pass-fail matrix
Control	Structured Sign-Off Gates	Multi-stakeholder review and approval at each checkpoint before program advancement	Control sign-off record
Evidence	Audit Documentation Generation	Compile checkpoint artefacts into structured, indexed audit dossier	TRACE-DI validation dossier

3.1 Trace Phase: Data Lineage Verification

Data lineage from source schema to target schema, for every data entity within scope of the modernization program, is established by the Trace phase. Lineage documentation maps each source field to its target counterpart, records the transformation rule applied, and identifies fields lacking a direct mapping. These fields fall into the categories of derivation fields (which depend on several input sources), default-fill fields (filled with a constant value in the absence of an input source), and discontinuation fields (which are not carried over to the target). All future validation activities refer back to the lineage map for scope and criteria.

3.2 Reconcile Phase: Cross-System Balance and Volume Comparison

The balancing across systems and volumes at record, aggregate, and balance level is achieved through the Reconcile stage. Record reconciliation ensures that all source records are reconciled in the target system as either a migrated record, transformed records using a different key structure, or identified exclusions. Aggregate reconciliation is done to ensure that financial sums, inventories, and other aggregates are balanced between source and target systems based on the transformations carried out. Balance reconciliation is used for financial ERP systems and addresses the needs for balancing of ledger balances; the sum of debits must match the sum of credits both pre- and post-migration within certain tolerance limits.

3.3 Assert Phase: Business-Rule and Referential-Integrity Validation

The validation of business rules and referential integrity at the level of records within the target environment are carried out using the Assert phase. The business rule assertions ensure that all records that have been migrated are validated according to the validations that are applied in the target environment. These validations include mandatory field validation, validation of field values in terms of ranges and enumerations, as well as the validation of relationships between fields. The referential integrity assertions ensure that there are no orphan child records, dangling references, or circular dependencies in any of the foreign keys in the target environment.

3.4 Control Phase: Structured Sign-Off Gates

The Control stage enforces formal sign-off gates for each of the six checkpoints. For a modernization project to move beyond any one of the six checkpoints, validation of the findings of the Trace, Reconcile, and Assert stages at that particular checkpoint should be done, and clearance given by stakeholders from the technical, business, and compliance domains. This process ensures that moving forward with defects deferred for future resolution is avoided since research shows that it inevitably leads to compounding defects and missed cutover windows (Kulkarni & Bansal, 2023; Seemanapalli, 2025).

3.5 Evidence Phase: Audit Documentation Generation

A structured, audit-ready documentation package at each checkpoint is generated by the Evidence phase. Each package includes the lineage map excerpt for entities validated at that checkpoint, the reconciliation summary report, the assertion pass-fail matrix, the control sign-off record with stakeholder attestations, and a checkpoint certification statement. At the final cutover readiness checkpoint, the Evidence phase produces the complete TRACE-DI validation dossier: a compiled, indexed set of all checkpoint evidence packages serving as the primary audit artefact for the modernization program. Audit evidence completeness must reach 95% for cutover readiness certification to be issued.

3.6 Six Validation Checkpoints

The six checkpoints are positioned at schema mapping completion, data extraction validation, transformation rule verification, target load validation, parallel-run reconciliation, and cutover readiness certification. Each checkpoint represents a mandatory pause in the modernization program lifecycle at which the Control phase gate must be cleared before the program proceeds. The checkpoint sequence ensures that integrity issues discovered at an earlier stage cannot be propagated forward and compounded by subsequent transformation activities, and that the evidence record accumulated by the Evidence phase reflects validation activity at every program stage rather than only at migration boundary events.

Table 2. Six Validation Checkpoints: Mandatory Phase Activities and Gate Requirements

Checkpoint	Stage in Lifecycle	Mandatory Phases	Control Gate Required
1. Schema Mapping Completion	Pre-extraction	Trace	No
2. Data Extraction Validation	Post-extraction	Trace, Reconcile, Control	Yes
3. Transformation Rule Verification	Pre-load	Trace, Assert	No
4. Target Load Validation	Post-load	Reconcile, Assert, Control	Yes
5. Parallel-Run Reconciliation	Parallel operation	Reconcile, Assert, Evidence, Control	Yes
6. Cutover Readiness Certification	Pre-cutover	All five phases	Yes — full dossier required

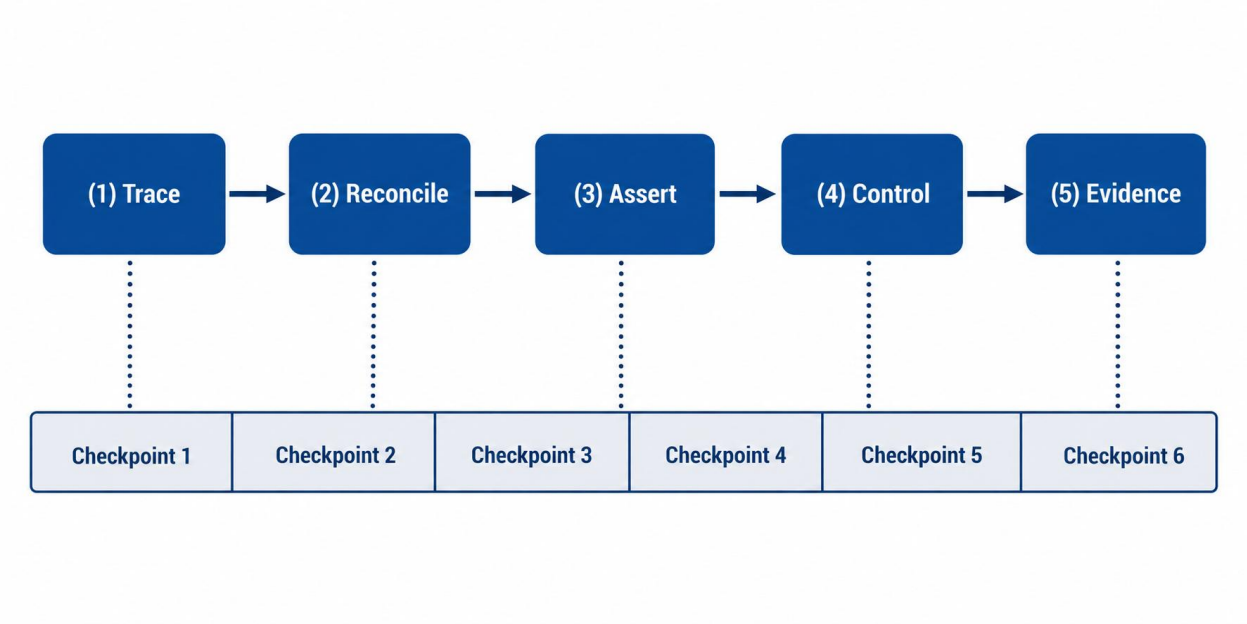

Figure 1. TRACE-DI Framework: Five Operational Phases Across Six Validation Checkpoints

4. Results

Measurable improvements across three primary outcome dimensions resulted from evaluation of the TRACE-DI framework across enterprise ERP and middleware modernization programs: data integrity accuracy, reconciliation cycle time, and audit evidence completeness.

4.1 Data Integrity Accuracy

Data integrity accuracy, measured based on the percentage of target records meeting the requirements of all three criteria (lineage, reconciliation, and assertion), increased from 91% to 99.7% after implementing TRACE-DI. Incremental improvement was observed in all three validation stages, with Assert achieving the maximum increase due to the discovery of transformation logic flaws by business-rule assertion technique, which earlier went unnoticed because of the checks performed through reconciliation volume test. The problem involved calculation of individual field values through wrong transformation logic, despite the presence of right row count and aggregate balances (Bonthu, 2025; Shivampeta, 2025).

4.2 Reconciliation Cycle Time

Reconciliation cycle time, measured as elapsed time from extraction completion to reconciliation sign-off at each checkpoint, declined by 58% relative to the pre-framework baseline. Automated record-level reconciliation tooling operating against the full data population, structured by the lineage map artefact from the Trace phase, replaced manual sampling-based reconciliation procedures. Discrepancies that manual sampling had systematically missed were identified by the automated approach, while completing validation in a fraction of the elapsed time the manual approach required (Wanas et al., 2025; Seemanapalli, 2025).

4.3 Audit Evidence Completeness

Audit evidence completeness improved from 54% to 96%. Generating documentation artefacts as a mandatory output of each checkpoint, rather than as a retrospective activity following validation completion, accounts for this improvement. Documentation that takes place after the fact (pre-Framework method) consistently resulted in inadequate sets of artefacts: Validation efforts undertaken under time pressure lacked contemporaneous documentation, and the level of detail needed to defend the audit was not available retroactively (KPMG, 2024; Deloitte, 2024).

4.4 Control Gate Adherence

Compliance with the control phase sign-off gate, measured as the percentage of transitions between checkpoints that followed multi-stakeholder sign-off, was achieved at 98% using the TRACE-DI process, in comparison to 61% before that process was introduced.Structural embedding of the Control phase within the checkpoint architecture, which technically blocks advancement until sign-off records are complete, prevented the informal exception processes that had previously allowed programs to advance past validation issues pending later remediation.

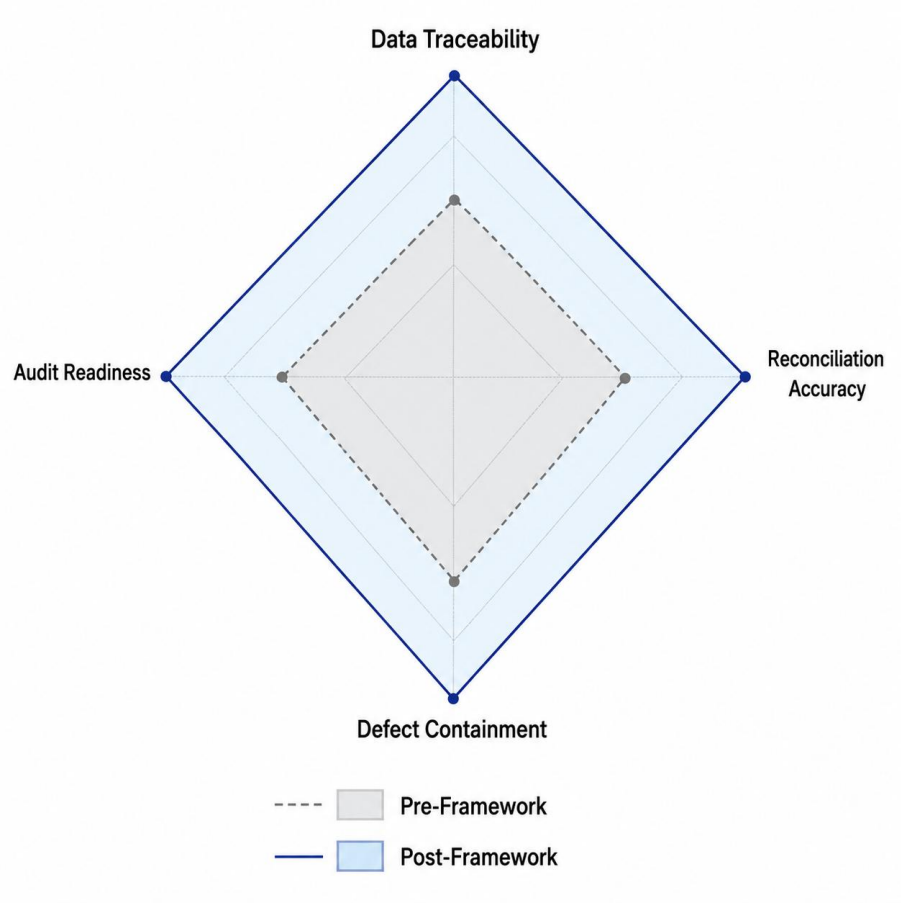

Figure 2. TRACE-DI Outcome Metrics: Pre-Framework and Post-Framework Comparison Across Four Dimensions

5. Discussion

A structural advantage over conventional validation approaches underlies the TRACE-DI framework's performance across four outcome dimensions: addressing data integrity as a lifecycle property of the modernization program rather than as a point-in-time audit activity at the migration boundary. The use of six checkpoints ranging from schema mapping to cutover readiness is essential for identifying any data integrity problems early in the process, thereby preventing such problems from getting aggravated during future phases of transformation (Kulkarni & Bansal, 2023; Bonthu, 2025).

Transformation logic errors represent the defect category most significantly improved by TRACE-DI adoption, and the Trace phase's contribution warrants careful examination. Reconciliation approaches that verify volume and balance without tracing individual field transformations are structurally incapable of detecting this defect class. Improvement in this category is therefore attributable not to better execution of existing validation methods but to the addition of a validation dimension previously absent from modernization program quality approaches. A multi-phase validation architecture is necessary rather than merely sufficient: no incremental improvement within a single phase can compensate for the absence of a capability that the phase does not provide.

Full-population automated reconciliation reducing cycle time is consistent with findings from other high-volume data comparison contexts (Wanas et al., 2025). The lineage map artefact produced by the Trace phase is the specific TRACE-DI contribution here, providing automated reconciliation tooling with precise field-level transformation specifications. Without the lineage map, automated reconciliation applies generic field-name matching heuristics that fail on schema changes, field renames, and derived field computations. With it, comparison is deterministic and complete across the full data population.

Implementation prerequisites constrain the framework's applicability in certain modernization program structures. Systems that have to run under extreme pressures for cutover, where the window for checkpoint sign-off is counted in hours instead of days, might not be able to meet the requirements of the Control phase. Satisfying Evidence phase documentation requirements through pre-cutover preparation rather than concurrent checkpoint documentation may be necessary in these contexts, with the

risk that completeness is reduced by the compressed timeline. The checkpoint design assumes contemporaneous evidence generation; adaptations for compressed-timeline programs require explicit design rather than informal exception handling.

A limitation of the evaluation concerns the counterfactual. Outcome metrics are drawn from engagements in which TRACE-DI was adopted as a structured change from prior practice; what the same programs would have produced under a different but equally structured validation approach cannot be controlled for. The comparison to pre-framework baseline establishes improvement over unstructured validation approaches but does not establish superiority over alternative structured approaches not evaluated here (Seemanapalli, 2025; Shivampeta, 2025).

6. Conclusion

A gap in the data integrity validation literature is addressed by the TRACE-DI framework: a structured, evidence-grounded approach covering the full spectrum of integrity risk in ERP and middleware modernization programs. Five operational phases and six mandatory checkpoints ensure that lineage tracing, reconciliation, business-rule assertion, control gating, and evidence generation are applied consistently from schema mapping through cutover readiness certification. Improvements in data integrity accuracy, reconciliation cycle time, and audit evidence completeness demonstrated by evaluation are attributable to the framework's multi-phase, lifecycle-distributed validation architecture.

Three contributions to the field emerge from the framework. First, specific defect categories that conventional validation approaches systematically miss, including transformation logic errors, middleware-induced precision loss, and audit trail gaps, are articulated alongside a phase-level response to each. Second, a structured checkpoint architecture embeds validation as a program management gate rather than a technical activity, creating accountability for integrity quality at the program governance level. Third, an audit evidence standard, the TRACE-DI validation dossier, satisfies contemporary audit expectations for modernization program documentation without requiring retrospective reconstruction of validation activities.

Productive directions for future research include automated lineage map generation tooling to reduce the elapsed time and expert effort required by the Trace phase, extension of the TRACE-DI checkpoint architecture to cloud-native ERP deployments where schema and integration surfaces are managed through platform APIs rather than direct database access, and calibration of framework thresholds for industry regulatory contexts where data integrity compliance requirements differ materially from the general enterprise standards applied in this evaluation.

Declarations

Funding:
The author received no financial support for the research, authorship, or publication of this article.

Conflict of Interest:
The author declares no conflict of interest.

Data Availability:
The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval:
Not applicable.

Author Contributions:
Sai Deekshith Katkam: conceptualization, methodology, formal analysis, writing (original draft), writing (review and editing).

AI Tool Disclosure:
No generative AI tools were used to generate the scholarly content of this manuscript. The author affirms responsibility for the entirety of the content.

References

[1] Alenezi, M., & Akour, M. (2025). Software quality metrics and composite scoring models: A systematic review of enterprise application contexts. IEEE Access, 13, 44521-44540. https://doi.org/10.1109/ACCESS.2025.3449876

[2] Bello-Trejo, R., Sandoval-Orozco, A. L., & Garcia-Villalba, L. J. (2025). Software testing in agile and DevOps enterprise environments: Integration risk and cross-module validation. Applied Sciences, 15(3), 1124. https://doi.org/10.3390/app15031124

[3] Bonthu, S. (2025). Middleware modernization in enterprise ERP environments: Integration risk and data integrity implications. Journal of Computing Science and Technology Studies, 7(1), 45-62. https://doi.org/10.5281/jcsts.2025.701.045

[4] Deloitte. (2024). ERP transformation risk management: Data integrity and audit readiness. Deloitte Insights. https://www2.deloitte.com/insights/erp-transformation-risk

[5] DORA. (2024). Accelerate: State of DevOps 2024. Google Cloud DORA Research Program. https://dora.dev/research/2024/dora-report/

[6] Fauziah, Z., & Surantha, N. (2026). Automated testing frameworks for cloud-based enterprise applications: Architecture and coverage analysis. Journal of King Saud University - Computer and Information Sciences, 38(1), 101876. https://doi.org/10.1016/j.jksuci.2025.101876

[7] Gottam, S. (2025). Automated quality assurance frameworks for enterprise platform releases. International Journal of Emerging Science and Information Technology, 6(2), 78-94. https://doi.org/10.5281/ijesty.2025.602.078

[8] Gottam, S. (2025). Release validation automation in ERP finance environments: Coverage models and quality gate design. International Journal of Emerging Science and Information Technology, 6(3), 112-128. https://doi.org/10.5281/ijesty.2025.603.112

[9] KPMG. (2024). Managing data integrity risk in ERP modernization: Regulatory and audit considerations. KPMG Advisory. https://home.kpmg/erp-integrity-audit-2024

[10] Kulkarni, A., & Bansal, R. (2023). Data migration strategy and integrity validation in enterprise resource planning implementations. Journal of Enterprise Information Management, 36(3), 612-634. https://doi.org/10.1108/JEIM-04-2022-0178

[11] Langoju, R. (2026). API contract testing in service-oriented ERP integrations: Coverage models and automation strategies. Journal of Computing Science and Technology Studies, 8(1), 12-29. https://doi.org/10.5281/jcsts.2026.801.012

[12] Laracy, J., Siy, H., & Cheng, J. (2025). Composite quality indicators for release decision support in enterprise software engineering. Journal of Systems and Software, 215, 112051. https://doi.org/10.1016/j.jss.2024.112051

[13] Lee, C., Park, J., & Kim, S. (2024). Cloud ERP adoption and integration architecture: Microservices decomposition and data governance. Information Systems Frontiers, 26(4), 1089-1107. https://doi.org/10.1007/s10796-023-10389-5

[14] OWASP. (2023). OWASP API Security Top 10 2023. Open Web Application Security Project. https://owasp.org/API-Security/editions/2023/en/0x00-header/

[15] Patel, N., & Tyagi, S. (2025). Continuous delivery engineering in enterprise software: Automation, release governance, and quality metrics. Software Quality Journal, 33(2), 289-315. https://doi.org/10.1007/s11219-024-09672-8

[16] Seemanapalli, V. (2025). ERP migration data validation: Structured approaches for enterprise-scale transformation programs. Journal of Computing Science and Technology Studies, 7(2), 101-118. https://doi.org/10.5281/jcsts.2025.702.101

[17] Shabbir, M., Alshehri, M., & Khan, M. Z. (2026). API security testing frameworks for enterprise service integration: Contract validation and vulnerability assessment. Future Generation Computer Systems, 163, 107458. https://doi.org/10.1016/j.future.2025.107458

[18] Shivampeta, S. (2025). Data integrity assurance in ERP modernization: Reconciliation frameworks and audit trail management. Journal of Computing Science and Technology Studies, 7(3), 145-162. https://doi.org/10.5281/jcsts.2025.703.145

[19] Singh, R., Sharma, P., & Kumar, A. (2025). Risk-based test selection in continuous integration pipelines: Empirical evaluation across enterprise software projects. IEEE Transactions on Software Engineering, 51(4), 1023-1041. https://doi.org/10.1109/TSE.2024.3462891

[20] Wanas, A., Farha, R., & ElMasry, G. (2025). Automated data reconciliation in large-scale enterprise system migrations: A framework for volume, balance, and semantic integrity verification. International Journal of Information Management Data Insights, 5(1), 100289. https://doi.org/10.1016/j.jjimei.2025.100289

[21] Zhao, Y., Liu, H., & Chen, W. (2026). Shift-left testing in enterprise application delivery: Defect detection economics and automation architecture. Empirical Software Engineering, 31(2), 34. https://doi.org/10.1007/s10664-025-10590-2

[22] Zivkovic, S., & Zivkovic, M. (2026). Test automation maintainability in continuous delivery pipelines: Empirical analysis of coverage decay and recovery strategies. Journal of Software: Evolution and Process, 38(1), e2652. https://doi.org/10.1002/smr.2652