Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Digital Twin Frameworks for Predictive Risk Management in ERP Transformations

Rajasekhar Reddy Putta*1, Srinivas Siruvari*2

Pondicherry University, India.

JNTU Hyderabad, India

ARTICLE INFO

Received: 08 April 2026

Accepted: 12 April 2026

ABSTRACT

ERP transformations are among the most complex and failure-prone enterprise transformations in modern enterprise life. They consistently show patterns of investment overruns, schedule slippages, and scope shortfalls that earlier risk management frameworks have not been able to manage. The Digital Twin for ERP Transformation framework combines system-of-systems modeling, streaming telemetry, and machine learning-based predictive risk analytics with existing project risk management methodologies to create a dynamic computationally synchronized digital twin of the transformation's risk profile. Guided by ISO 23247 digital twin architecture, NIST AI RMF governance function, and PMBOK risk management processes, operationalize a four-layer digital twin reference architecture of observable program components, integration and telemetry, twin core models, and decision applications. Engineered leading indicators, including change-request entropy, defect acceleration index, backlog flow efficiency, and migration yield, are fed to machine learning classifiers and Monte Carlo simulation engines. This replaces milestone plans with honest, continuously updated uncertainty distributions, which are used to predict probabilistic schedule and cost risk curves. Pathway-specific calibration for greenfield, brownfield, and hybrid ERP migration strategies requires different risk profiles and governance requirements. A system of explainability, bias testing, and human-in-the-loop accountability structures for AI systems to meet AI trustworthiness principles is used to achieve governance obligations. The resulting program risk management framework changes the model of status reporting from after-the-fact to continuous, predictive, actionable intelligence for low-cost intervention before risk materializes, considerably lowering the risk of program failure in complex multi-phase ERP transformation projects.

Keywords: Digital Twin, ERP Transition, Predictive Risk Analysis, Machine Learning, Artificial Intelligence Governance

1. Introduction

Enterprise-wide ERP implementations are rated consistently and widely as the most important, the riskiest, and the most failure-prone category of projects that enterprises engage in. It is an established industry fact that an enormous majority (55-75%) of ERP implementations do not achieve their business objectives across all industries and company sizes [1]. In a large sample of diverse IT projects, the major contributors to failure were weak change management, data quality problems, lack of executive support, and scope underestimation, which mutually influence each other in ways not amenable to prediction from static risk registers [1]. The financial impact for organizations can be massive. Reports from consulting organizations estimate the global cost to organizations of failing to deliver these ERP transformations at several trillion dollars due to delayed benefits, rework, and operation disruption.

Each dynamic share of risk has a number of elements of uncertainty regarding the redesign process, data quality and migration, integration risk, the configuration's stability, and the user's and business

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

owner's acceptance of organizational change. This risk landscape is structurally opposed to many standard methods of risk management in projects, including looking at the risk register at regular intervals, RAG status reporting, and milestone-based escalation. By contrast, these are retrospective, capturing what has gone wrong, not what is going to go wrong. By the time a risk comes to attention via a status report, the window for low-cost fixes that have the most impact has usually closed.)

Digital twins have their origins in manufacturing and industrial processes, where they maintain a digital copy of a physical system in the digital world and update it with streaming telemetry. Rather than only being used for reporting, they can be used to compute scenarios and predictions and to optimize systems forward in time. Applied to an ERP transformation program, this framework can model the full socio-technical program, including requirements and configurations, test iterations, data migration pipelines, environment health, and organizational adoption signals. This turns project risk management from a postmortem data analysis exercise into an always-on, near-real-time process. The Digital Twin for ERP Transformation (DT-ERPT) integrates system-of-systems modeling, streaming telemetry monitoring, and machine learning predictive risk analytics with projects' best-practice risk management processes to create a computational model of qualitative and quantitative risk processes that is always on. Despite continuous reporting of poor overall success rates in complex IT projects, large interrelated and interdependent projects are more likely to fail than smaller ones, revealing the need for a more analytical and proactive approach to program risk management [2].

2. Reference architecture and conceptual foundation

2.1 From Dashboard to Digital Twin

The difference between a program dashboard and a digital twin is more than merely a visual one. It is an architectural and functional difference at every layer of the overall system architecture. A dashboard describes what has happened. It is simply a snapshot of measurements compiled to the present time (limited by the lag of reporting), of use mainly for retrospective analysis. The prescriptive, generative nature of the ERP transformation digital twin can help answer what-if questions: For example, what is the new P80 completion date if twelve percent of test capacity is reallocated to integration automation, or what is the cutover risk if three high-blast-radius transports are resequenced? The output of the twin is a leading indicator of risk that appears days or weeks beforehand, prior to the enterprise experiencing negative outcomes. AI risk management frameworks use the terms "explainability," "accountability," and "continuous monitoring." It is particularly applicable to the digital twin in a program risk context, in which the model output becomes the basis for real decisions on resourcing and scheduling [3]. The digital twin recomputes probabilistic completion curves as new telemetry is received. The end result is that the milestone plans are replaced with accurate uncertainty distributions, which describe the state of the program at any moment rather than how it was projected months before.

This differs from the more general definition of "digital twin" in manufacturing and networks, where the aim is not simply to visualize the system but also to analyze and simulate its behavior, e.g., by scheduling maintenance actions and capacity plans or predicting failures well before they manifest themselves. The same principle applies to an ERP program as a complex system whose internal state can be monitored, predicted, and guided to prevent escalation towards an undesirable situation. This argumentation is also observable in the ISO 23247 digital twin framework for manufacturing, which specifies a layered architecture consisting of observable entities, device communication interfaces, digital twin entities, and user entities that jointly maintain a synchronous view of an asset (or process) for analysis, prediction, and control purposes [4]. The same architecture with its synchronous view principle between the real and digital world can be applied to set up the ERP program in our case,

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

when program artifacts (requirements, transports, test cases, migration objects, and environments) substitute manufacturing assets .

2.2 Layered Reference Architecture

The reference architecture adds four interconnected layers on top of the ISO 23247 layered architecture. The top layer contains observable (to the twin) program elements. They include requirements and backlog items, configuration transport logs, code repository events, continuous integration and deployment results, data migration ETL logs, test case execution and defect lifecycle events, environment uptime and incident events, service desk tickets and non-technical stakeholder comms sentiment, training completion and quality index scoring, financial burn against plan, and vendor service-level events. The second tier is integration and telemetry: APIs, event collectors, log shipping pipelines, and a message bus that integrates signals into a common risk ontology. The semantic integration allows structural dependency analysis (to allow risk prediction). The third tier, the twin core, implements the structural dependency graph of artifacts, the time-indexed state store, and a set of risk models (classification, regression, survival analysis, and Bayesian networks modeling causal relationships), plus the simulation engine (supporting Monte Carlo analysis of schedule and cost risk and discrete-event simulation of cutover wave planning). The fourth tier features decision applications, including a PMO risk cockpit, probabilistic release readiness indicators, a change-impact explorer, a next-best mitigation recommender, and a scenario planner to assess portfolios of possible mitigations with respect to their tail risk and cost.

Architecture Layer	Primary Components	Risk Management Function
Observable Program Elements	Backlog, transports, test cases, migration logs, environment health, sentiment signals, financial burn	Defines the observable universe of program telemetry feeding the twin
Integration and Telemetry	APIs, event collectors, log shipping, message bus, semantic ontology mapping	Harmonizes disparate tool data into a unified, queryable risk data fabric
Twin Core	Dependency graph, state store, ML risk models, Bayesian networks, Monte Carlo engine	Computes leading indicators, probabilistic risk scores, and what-if simulations
Decision Applications	PMO cockpit, readiness gates, change-impact explorer, mitigation recommender, scenario planner	Translates twin outputs into actionable program decisions and governance artifacts

Table 1: Layered Reference Architecture - Components and Functions [3, 4]

3. Predictive Risk Modeling: Leading Indicators and Quantitative Methods

3.1 Engineered Leading Indicators

The quality of the model depends on leading indicators, which are features derived from telemetry data that predict future program risk rather than describe its current state. The model is implemented using four leading indicators, which have been evaluated to show their early warning properties in the complex program environment. Change-request entropy is the Shannon entropy of the change-request topic distribution of change requests opened in an observation window. Change-request entropy has been shown to predict schedule regression due to scope creep, rework cycles, and design instability. A measure of change-request entropy detects change early in the milestone tracking process, before any milestone effects happen. . The defect acceleration index indicates the degree of change in the defect discovery rate, factoring in the coverage of testing, as an indicator of the length of the stabilization tail of the stabilization period for the system integration test phase. Backlog aging and flow efficiency measure the value-adding cycle time divided by the total time since the work items

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

were created. A drop in flow efficiency is a predictor of cycle time blowouts in later delivery periods. Migration yield and reconciliation error rates (percentages of data objects successfully migrated and reconciled against source systems per conversion run) provide advanced indicators of data readiness risk, several months ahead of planned cutover rehearsals.

Therefore, the leading indicators monitor the areas that seem to be the main causes of ERP project failures: data, testing, organizational change management, and scoping [1]. The knowledge that these failure modes are generally predictable and preventable is what drives the move towards engineered predictive signals instead of passive risk registers. These risk factors are given to the machine learning classifiers and regression models as time series-related features. The risk modeling pipeline, including feature engineering, model training and validation, model deployment, and drift monitoring, is the analytical engine at the heart of the twin core and is what sets it apart from a normal program analytics framework.

3.2 Quantitative Risk: Monte Carlo and Bayesian Methods

Quantitative risk analysis for the twin core adopts two related models. The Monte Carlo simulation of potential events entering the program is based on aleatory uncertainty in the total program network, with parameterization based on empirical posterior distribution functions over all program data. Epistemic uncertainty is managed by constantly updating the posterior distributions when the twin observes the state of the program. The outputs are probability distribution curves of P50 and P80 estimates of completion time and cost consumption. This replaces the deterministic precision of a schedule with explicit uncertainty ranges and supports the risk-management process. PMBOK distinguishes qualitative risk analysis from quantitative risk analysis as two sequential and complementary processes. Quantitative analysis enables the determination of the probabilistic basis for risk response planning and the specification of contingency reserves through Monte Carlo and sensitivity analysis [9]. Bayesian networks allow the encoding of causal assumptions through expert priors. For example, there may be a known causal relationship between how flaky the environment is and how high the test rework rate is. These properties can be raised in situ via telemetry, making risk scores interrogable during steering committee meetings, not black boxes.

Leading Indicator	Measurement Basis	Signal Type	Primary Risk Domain
Change-Request Entropy	Shannon entropy of CR topic distributions over observation window	Forward-looking volatility signal	Scope, design stability, rework
Defect Acceleration Index	Rate of change of defect discovery adjusted for active test coverage	Stabilization lag predictor	Quality, test cycle closure
Backlog Aging and Flow Efficiency	Ratio of value-adding cycle time to total elapsed wait time	Cycle-time blowout predictor	Schedule, delivery throughput
Migration Yield and Reconciliation Error	Percentage of data objects successfully migrated and reconciled per run	Data readiness early warning	Data migration, cutover viability

Table 2: Leading Indicators - Definition, Signal Type, and Risk Domain [5, 6]

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4. Align within ERP transformation pathways

4.1 Pathway-Specific Risk Profiles

There are three types of migration to new systems in enterprise resource planning. Each has a different risk profile and thus a different methodology to use to calibrate the twin to gather risk intelligence. Greenfield, which builds a new configuration mostly using standard processes, is a riskier implementation in terms of process redesign. For organizations taking this route, there are hundreds of fit-to-standard decisions to be made. The indicator for the risk of downstream delivery is the maturity of the blueprint in the sense of how stable, tested, and adopted across the business are the process designs. The twin in a greenfield setting is blueprint maturity, fit to standard variance, and ready to enable adoption of change. This provides the PMO visibility of potential design instabilities before configuration and testing.

Studies of SAP S/4HANA adoption patterns indicate that nearly half of all respondents have managed to deliver on their transformation plans ahead of the deadline SAP had set in 2027 to stop providing support for legacy ERP software, though many organizations also report that they have experienced cost and time overruns, creating a gap between how much progress is made and how well execution happens. In a brownfield migration, all configurations, customizations, and data in the source system are preserved. . The quantity of custom code that needs remediation, the throughput of data conversion runs, and the regression defect rate during business testing are important brownfield program metrics. The twin weights these metrics to track custom-code remediation progress against planned throughput curves so SAP can determine that the technical debt of the legacy system is remediated on time for the go-live gates. SAP's selective data transition engagement model addresses the third pathway, hybrid migrations. The selective data transition approach is a means for organizations to migrate selective data objects, not whole datasets. Such an approach comes with a specific risk profile with cutover graph dependencies, interface harmonization requirements and data lineage across the transition boundary [6].

4.2 Illustrative Risk Scenario

The ground this in the context of a multi-country hybrid migration program with over 1200 interfaces, and the change-request entropy and defect acceleration index are trending upwards at an accelerating rate through a second system integration test cycle, the predicted probability of being beyond the integration gate in five weeks is sixty-two percent. Scenario analysis in the twin yields three mitigations to pursue: (1) devoting 12 percent of developed capacity to integration test automation, (2) rescheduling the three high-blast-radius transports to reduce the number of dependent conflicts, and (3) pushing a nightly data reconciliation control upstream to catch migration exceptions earlier in the conversion process. Simulating the mitigation portfolio yields an overall P80 lateness estimate reduction of 3.4 weeks for less than 1.2% of the program budget, an outstanding ROI according to the risk-adjusted analysis. The analysis shows how the twin can change risk management from an administrative activity into a decision-supporting framework for assessing the value of particular mitigations, rather than only requiring the documentation and assignment of mitigation ownership.

Transformation Pathway	Dominant Risk Profile	Twin Emphasis	Key Leading Indicators
Greenfield	Process redesign and blueprint instability	Blueprint maturity indices, fit-to-standard variance	Change-request entropy, adoption readiness metrics
Brownfield	Migration compatibility and technical debt	Custom code remediation throughput, regression defect kinetics	Defect acceleration index, migration yield

Journal of Information Systems Engineering and Management
2026, 11(3s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Hybrid / Selective Data Transition	Cutover complexity and interface harmonization	Cutover graph dependencies, data lineage integrity	Reconciliation error rates, interface test coverage

Table 3: ERP Transformation Pathway Risk Profiles and Twin Calibration [7, 8]

5. Governance, Ethics, and Operational Sustainability

5.1 AI Governance and Trustworthiness

The implementation of machine learning models in a program risk context creates governance requirements that go beyond customary program governance, as those requirements need to be designed into the program structures from the beginning. The NIST AI Risk Management Framework for AI uses four governance functions (GOVERN, MAP, MEASURE, and MANAGE) as a lifecycle approach to the responsible deployment of AI [3]. The GOVERN function guides models and the system by identifying the policies, roles, and accountabilities that frame a model's ownership of and actions on its outputs and the escalation of critical risk conditions surfaced by the twin. The MAP function places the AI model in the broader system, defining the potential harms of false negatives (when conditions amass and problems go unaddressed) and false positives (when incorrect alerts prompt costly and unnecessary interventions that erode stakeholder trust in the model). The MEASURE function computes the area under the precision-recall curve for risk flag classification, the Brier score for late and overrun risks probabilities, precision and recall rates for alerts, the frequency of drift events, and time-to-retrain as a health signal for the models.

For this governance component, explainability is non-negotiable. Feature attribution techniques are embedded in PMO cockpit views and steering committee presentations: risk scores can be queried and challenged, preventing them from being treated as inscrutable outputs of an unwieldy system. Bias testing across workstreams and vendor teams gauges where noisier telemetry may make the program appear riskier than its peers due to the properties of the data (for example, the level of tooling integration or the quality of data entry by personnel). Governance for software and systems engineering for complex systems increasingly includes requirements traceability and accountability structures to hold AI-assisted decision systems accountable to organizational goals and ethical constraints across the system lifecycle [7].

5.2 Human-in-the-Loop and MLOps Sustainability

The twin proposes; human decision-makers dispose. Automating evidence acquisition and risk scoring does not transfer accountability from program leaders to algorithms. Rather, it increases the quality, timeliness, and rigor of human judgment by ensuring that decision-makers are not relying on intuition and incomplete status reports but on the best evidence available. Risk response decisions, including the mitigation portfolios selected in the scenario engine, will still be subject to human scrutiny and approval and consistent with standard project governance mechanisms. The PMBOK Guide framework for risk management, Plan, Identify, Qualitative Analysis, Quantitative Analysis, Plan Responses, Implement, and Monitor, is the procedural scaffolding that the twin is built on. The twin is evidence-based and automated in its evidence-acquisition dimension, but remains human-in-the-loop in the decision-making dimension [9].

Model fidelity is maintained through a disciplined MLOps operating model, with model maintenance as an active operational responsibility. Telemetry is monitored with data quality guards to watch for outliers or schema drift in incoming streaming inference telemetry. Concept drift detection identifies changes in underlying statistical relationships in the model, which always occur through model design, build, test, and cutover, triggering automated retraining workflows with holdout evaluation before versions can be promoted to production. Model scorecards and governance audit logs provide evidentiary trails for mandated AI model governance reviews. Analysis of industry data on ERP

projects found that the most common causes for project schedule and cost overruns were lack of project monitoring and project course corrections, providing a compelling operational case for replacing periodic review cycles with continuous model maintenance [10].

AI RMF Function	Governance Obligation	DT-ERPT Operationalization	Accountability Owner
GOVERN	Policies, roles, and accountability structures for AI deployment	Model ownership definitions, escalation paths, decision authorization protocols	Program Steering Committee, PMO
MAP	Context and harm identification for AI application	False negative and false positive impact assessment, risk taxonomy alignment	Risk Lead, AI Governance Officer
MEASURE	Operational metrics for model performance and trustworthiness	Precision-recall scores, Brier calibration, drift frequency, time-to-retrain tracking	MLOps Team, Model Governance Board
MANAGE	Controls and response mechanisms for identified AI risks	Retraining workflows, bias remediation, human-in-the-loop authorization gates	PMO, Release Management, AI Owner

Table 4: NIST AI RMF Governance Functions Mapped to DT-ERPT Operations [9, 10]

Conclusion

Although ERP transformations will always be among the most consequential and difficult planned investments that organizations undertake, the universal experience in the industry of cost overruns, schedule overruns, and scope shortfalls indicates that there is something fundamentally flawed about the current model of program risk management. The Digital Twin for ERP Transformation framework addresses that flaw by replacing slow, retrospective, static risk registers with a continuously synchronized computational model of the evolving state of an ERP transformation program, generating and continuously updating probabilistic risk intelligence from live telemetry data across all sources of risk: requirements, configuration, testing, data migration, environments, and organizational adoption signals. Combined with the ISO digital twin architecture, the NIST AI governance, and the PMBOK risk management lifecycle model, this gives the framework predictive capability along with accountability, explainability, and human oversight, so that no important program decision is left to the opaque output of an algorithm. Combined with leading indicators that predict risk failure modes; Monte Carlo simulation replacing false milestone precision with honest uncertainty distributions; and pathway-specific calibration for greenfield, brownfield, and hybrid migration strategies, this is a new level of program oversight capability. As ERP systems become more complex, and the organizational investment in transformation programs increases, the focus of risk intelligence is shifting to predictive, real-time, and governance-based frameworks. A systems-based approach to risk intelligence is no longer simply an enhancement; it is a requirement for any organization which is serious about ensuring that the value of transformation programs is protected .

References

[1] NIST, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

Research Article

[2] Jeffrey K. Pinto and Dennis P. Slevin, "Critical factors in successful project implementation," IEEE Transactions on Engineering Management, 1987. [Online]. Available: https://ieeexplore.ieee.org/document/6498856

[3] Antonio Vetrò et al., "Open data quality measurement framework: Definition and application to Open Government Data," Government Information Quarterly, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0740624X16300132?via%3Dihub

[4] International Standardization, "Automation systems and integration — Digital twin framework for manufacturing —," 2021. [Online]. Available: https://cdn.standards.iteh.ai/samples/75066/ec0a1c59176e488887873acda6b7ecd9/ISO-23247-1-2021.pdf

[5] Blake Baltazar, "The State of SAP S/4HANA Adoption: Trends, Successes, and Challenges," ASUG Research, 2024. [Online]. Available: https://www.asug.com/insights/the-state-of-sap-s-4hana-adoption-trends-successes-and-challenges

[6] SAP SE, "SAP S/4HANA Selective Data Transition Engagement." [Online]. Available: https://support.sap.com/en/offerings-programs/support-services/data-management-landscape-transformation/selective-data-transition-engagement.html

[7] IEEE, "29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," 2018. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.1201/9781003129226

[8] Godlan Learning Center, "ERP Implementation Failure Statistics: 2025 Research." [Online]. Available: https://godlan.com/erp-implementation-failure-statistics/

[9] Project Management Institute, "PMBOK® Guide," 7th ed., PMI, Newtown Square, PA, 2021. [Online]. Available: https://www.pmi.org/standards/pmbok

[10] Lisa Schwarz, "10 Reasons for ERP Failures and How to Avoid Them," Oracle NetSuite, 2024. [Online]. Available: https://www.netsuite.com/portal/resource/articles/erp/erp-failure.shtml