

Predictive Risk Management in Large-Scale Technology Programs: A Data Analytics Framework for Early Risk Detection and Decision Support

Ayokunmi Paul Sodamola [1*], Mercy Osato Omoifo-Irefo [2], Chiamaka Beauty Obioha [3]
[1] Department of Management Information Systems, Baylor University, Texas, USA
[2] Department of Environmental and Public Health Sciences, University of Cincinnati, Ohio, USA
[3] Department of Accountancy, Yaba College of Technology University, Lagos, Nigeria

* Corresponding Author: **Ayokunmi Paul Sodamola**

Article Info

P-ISSN: 3051-3383
E-ISSN: 3051-3391
Volume: 06
Issue: 01
Received: 08-12-2024
Accepted: 06-01-2025
Published: 04-02-2025
Page No: 35-44

Abstract
Large-scale technology programs operate under persistent uncertainty arising from schedule interdependencies, cost pressure, technical complexity, changing requirements, governance constraints, and the coordination of multiple projects and stakeholders. Conventional risk-management processes remain indispensable, but periodic reviews and qualitative assessments may not fully exploit the time-stamped operational data generated throughout program execution. This study develops a longitudinal predictive risk-management framework designed to identify adverse technology-program conditions before formal risk escalation and translate predictive information into managerial decision support. The proposed empirical setting consists of major U.S. federal IT investments observed through 2024. The unit of analysis is the investment-reporting period. The primary outcome is a subsequent adverse CIO risk condition within a prespecified prediction horizon, with rating deterioration and severe cost or schedule variance evaluated as secondary outcomes. Predictor variables include current and lagged cost and schedule variance, temporal trends, volatility, spending dynamics, project concentration, prior risk trajectory, and scale indicators. Logistic regression, decision trees, random forests, and gradient boosting are evaluated using chronological validation and an independent out-of-time test set. Performance criteria include recall, precision, F1-score, ROC-AUC, PR-AUC, Brier score, calibration, and warning lead time. A real-data feasibility pilot of 50 quality-screened completed project records shows meaningful variation in cost and schedule performance while revealing the importance of explicit data-quality controls. Final predictive performance is intentionally not fabricated and must be populated after execution of the full historical panel. The framework contributes a replicable architecture connecting operational program signals, predictive risk intelligence, explainable model outputs, and human-governed management interventions.

DOI: https://doi.org/10.54660/IJAIET.2025.6.1.35-44

Keywords: Predictive Risk Management, Technology Programs, Project Risk, Data Analytics, Machine Learning, Early Warning, Program Management, Decision Support

Introduction

Large-scale technology initiatives combine technical development with organizational change, coordination, contracting, governance, financial control, and the management of interdependent work. Early information-systems research established that software-development risk is multidimensional rather than reducible to a single probability-impact score [1]. International Delphi research subsequently identified a broad set of software-project risks recognized by experienced practitioners [2], while

The consequences remain material in contemporary large technology environments. In its assessment of 25 major Department of Defense IT business programs, the U.S. Government Accountability Office reported that 16 programs had experienced cost or schedule changes since January 2021. Twelve reported cost increases, and twelve reported schedule delays, with reported delays ranging from 3 to 33 months [5]. Longitudinal academic analysis of federal technology projects has likewise found meaningful relationships between project risks, process maturity, and performance using 82 projects observed over 519 quarterly periods [6]. These findings demonstrate both the substantive importance of technology-program risk and the feasibility of analyzing risk as a time-varying empirical phenomenon [5, 6].

The program context creates additional analytical requirements. Portfolio risk-management research indicates that risk transparency, formalization, monitoring, integration, and risk-coping capacity influence portfolio outcomes [7]. Research on uncertainty in project portfolios further demonstrates that uncertainty can originate from projects, organizations, and external environments and is interpreted through managerial framing and control mechanisms [8]. Program-level risk research has shown that coordinated sets of projects create delivery risks not captured adequately by isolated single-project analysis [9]. Recent work on project scale also cautions that "large-scale" should not be treated as a self-evident category because financial size, complexity, actors, time, and organizational reach can represent different dimensions of scale [10]. Accordingly, this study treats major federal IT investments as an empirical proxy for large-scale technology programs and tests scale-related features explicitly rather than assuming that budget alone captures program scale [7-10].

Traditional technology-risk management has considerable value. Risk identification, expert judgment, structured reviews, risk registers, formalized processes, contingency estimates, and managerial attention all contribute information that cannot be replaced simply by an algorithm [1, 2, 4]. Research also shows that users and project managers can perceive project risk differently [11], that IT budget overruns have identifiable project-level determinants [12], and that the effect of planning can depend on the level of project risk [13]. The limitation addressed here is therefore not that traditional risk management is useless. It is that much of it is periodic, judgment-centered, and contemporaneous, whereas the operational record of a technology program is longitudinal and can potentially support explicit forecasting [11-13].

Data-driven approaches provide a route from monitoring to prediction. Earlier quantitative work developed contingency-estimation models for software projects [14]. More recent software-risk research has used historical project characteristics and artificial neural networks to estimate project risk [15]. Machine-learning research in complex infrastructure has further modeled interactions among risks and systemic project effects rather than treating each risk factor independently [16]. These studies show that predictive modeling is plausible, but they do not eliminate critical methodological questions concerning temporal leakage, calibration, out-of-time generalization, interpretability, rare adverse outcomes, and the translation of probability estimates into actual governance actions [14-16].

The central research gap is therefore an integration gap between dynamic technology-program data, temporally valid risk forecasting, explainable prediction, and management intervention. Existing project-risk scholarship provides mature taxonomies and management principles [1-4, 7-9]. Predictive modeling provides algorithms capable of learning nonlinear relationships [15-18]. Yet a management-relevant contribution requires these streams to be joined into one system in which data available at time t estimate an adverse condition after t, performance is verified on later unseen periods, explanations identify the drivers of risk, and governance rules determine what a manager should do with the prediction.

The aim of this study is therefore to develop and empirically evaluate a data-analytics-based predictive risk-management framework for identifying emerging risks in large-scale technology programs and supporting proactive managerial decision-making.



Fig 1: Predictive Risk-Management Data flow

The study has seven objectives: identify program variables associated with adverse future conditions; determine which variables contain useful early-warning information; engineer temporal indicators of deterioration and volatility; compare interpretable and nonlinear prediction models; validate discrimination and calibration on temporally later data; explain the principal drivers of high-risk predictions; and translate predicted risk into a human-governed intervention framework.

The research questions are as follows.

RQ1: Which observable program-performance indicators are most strongly associated with subsequent adverse technology-program risk conditions?

RQ2: To what extent can historical and operational data predict risk escalation within a fixed forward horizon?

RQ3: Do temporal trend and volatility features improve prediction beyond contemporaneous indicators?

RQ4: Do nonlinear ensemble models provide materially better out-of-time predictive performance than an interpretable logistic baseline?

RQ5: How can calibrated risk probabilities and explanations be converted into practical governance interventions?

The associated hypotheses are:

H1, deterioration in schedule-performance indicators is positively associated with the probability of a subsequent adverse risk condition.

H2, deterioration in cost-performance indicators is positively associated with subsequent adverse risk conditions.

H3, temporal trend and volatility features provide incremental predictive value over current-period levels.

H4, program-scale and project-concentration indicators provide incremental predictive information after controlling for contemporaneous variance measures.

H5, at least one nonlinear ensemble model achieves greater out-of-time PR-AUC than the logistic-regression baseline.

H6, the final predictive model produces a positive median warning lead time before formal adverse CIO risk escalation.

The proposed contribution is threefold. Theoretically, it extends the view of technology-program risk from a periodically assessed managerial construct to a partially observable dynamic condition. Methodologically, it develops a leakage-controlled longitudinal forecasting design rather than a cross-sectional classification exercise. Practically, it connects calibrated risk probabilities and model explanations to explicit management interventions rather than stopping at algorithmic accuracy.

The study is delimited to technology investments for which sufficiently reliable historical observations can be obtained through December 31, 2024. It does not claim that federal IT investments are identical to all private-sector technology programs. It also does not interpret predictive associations as causal effects.

Literature Review and Theoretical Foundation

A project, program, and portfolio represent distinct levels of organizing. The distinction matters because risks can arise not only from activities inside one project but also from interactions among projects competing for shared resources or depending on common components. Portfolio research has shown that risk identification and formalization increase transparency and that risk monitoring and integration contribute to risk-coping capacity [7]. Program-risk research likewise demonstrates that coordinated project structures create distinct delivery-risk questions [9]. These findings justify moving the analytical unit above the isolated task or software module when studying large technology transformations [7, 9].

Technology-program risk can be organized into several analytically useful categories. Schedule risk concerns slippage, milestone delay, dependency delay, and deteriorating completion forecasts. Cost risk concerns deviation between planned and forecast or actual cost. Technical risk includes architecture, integration, performance, technology novelty, and implementation uncertainty. Scope risk arises from unstable requirements and change. Resource risk concerns staffing, skill availability, turnover, and utilization. Vendor risk concerns contracting and supplier performance. Governance risk concerns decisions, escalation, oversight, and stakeholder alignment. These categories overlap rather than operating independently. The sociotechnical model of Wallace, Keil, and Rai provides empirical support for linked social, technical, management, and performance risks [3], while analyses of software-project failure identify interconnected chains of failure causes rather than isolated drivers [24].

Complexity intensifies these interactions. Research on team-level absorptive capacity found a positive association between project complexity and delays and overspending, while team capabilities could only partially offset this effect [22]. Work on complex-project management similarly argues that some complex projects cannot be completely specified in advance and therefore require continuous problem solving and learning [21]. Mental-construal research involving 805 IT project managers further shows that the way managers mentally represent risk affects risk identification, impact assessment, response planning, and willingness to enact responses [23]. Risk management is therefore simultaneously a technical-information problem and a human judgment problem [21-23].



Fig 2: Evolution of risk management

This study adopts a sociotechnical systems perspective as its principal theoretical lens. Sociotechnical reasoning is appropriate because a technology program contains coupled technical and social-management subsystems, and deterioration in one part can propagate through others [3]. The predictive analytics system is not conceptualized as a replacement for this human system. It is an information-processing augmentation. Operational traces increase observability, models integrate dispersed signals, explanations expose the basis of a risk estimate, and program managers provide contextual judgment and select interventions.

This theoretical positioning also clarifies why predictive risk management is different from merely automating a probability-impact matrix. Reactive management responds after adverse performance has materialized. Proactive management identifies anticipated threats through planning, expertise, workshops, scenarios, and monitoring. Predictive management uses historical and current measurements to estimate a future outcome at a defined horizon. Prescriptive management goes one stage further by mapping predicted states to recommended actions. ISO 31000:2018 provides a broader risk-management framework centered on identifying, analyzing, evaluating, treating, monitoring, and communicating risk [25]. The proposed framework sits within, rather than outside, that risk-governance cycle.

The transition to prediction requires a clear distinction among analytical functions. Descriptive analytics asks what occurred. Diagnostic analytics investigates why an observed condition may have occurred. Predictive analytics estimates what is likely to occur next. Prescriptive analytics evaluates what intervention should follow. The present study is primarily predictive, but its decision-support architecture creates a controlled interface to prescriptive managerial action. It deliberately does not allow the model itself to make high-consequence program-management decisions.

The most promising predictors emerge from the intersection of prior risk theory and data availability. Schedule indicators include current schedule variance, absolute variance magnitude, rolling variance slope, milestone slippage, and acceleration of delay. Cost indicators include current cost variance, absolute variance, rate of change, and spending deviation. Scale indicators include number of active projects, cumulative project cost, concentration of spending, and project-duration structure. Governance and organizational extensions can incorporate unresolved issues, approval latency, risk-register growth, staff turnover, supplier performance, and critical-defect backlogs when proprietary enterprise data are available. Federal Dashboard data provide a public core, while richer organizational sources constitute an extension layer. The distinction prevents the methodology from claiming that variables absent from the public data were actually observed.

Predictive algorithms should be compared rather than selected in advance. Logistic regression provides coefficients, probability estimates, and an intelligible benchmark. Decision trees capture simple nonlinear thresholds but can be unstable. Random forests reduce variance by aggregating randomized trees [17]. Gradient boosting fits additive learners sequentially to reduce prediction error [18]. The comparison therefore tests whether additional nonlinear flexibility produces material out-of-time gains rather than assuming that machine learning must outperform a statistical baseline [17, 18].

Evaluation must reflect the decision problem. Accuracy can be misleading when adverse outcomes are uncommon. Precision-recall analysis is particularly informative for imbalanced binary classification because it focuses directly on performance among positive predictions [19]. Calibration also matters because a governance system may act differently on a predicted risk of 0.30 than 0.80. Finally, explainability matters because managers need to know which features are driving an intervention. SHAP provides a principled feature-attribution framework for interpreting individual model predictions [20].

The resulting research gap can now be stated precisely. Risk-management literature explains what project and program risks are and why management processes matter [1-13]. Predictive and machine-learning literature provides candidate modeling techniques [14-20]. Complex-project research emphasizes learning, information needs, scale, and managerial interpretation [10, 21-24]. What remains underdeveloped is a replicable longitudinal architecture that predicts a clearly defined future risk state, prevents future information from entering training features, tests performance on later periods, quantifies warning lead time, explains individual predictions, and connects those explanations to auditable human decisions.

Research Methodology

The research adopts a post-positivist quantitative orientation. The purpose is not to claim deterministic laws of program failure, but to evaluate whether observable program data contain reproducible probabilistic information about later adverse conditions. A retrospective longitudinal design is preferable to a cross-sectional design because early warning requires temporal ordering. Federal technology-project research has demonstrated the feasibility of time-series project analysis, while the Federal IT Dashboard provides current access to historical CIO ratings and project variance information [6].

The proposed study population consists of major U.S. federal IT investments with usable historical records during the observation window ending December 31, 2024. The empirical unit is the investment-reporting period, not the individual project. Projects nested within each investment generate lower-level features that are aggregated to the investment-period level. This structure fits the conceptual focus on programs while recognizing that a federal “major IT investment” is an operational proxy, not necessarily an exact one-to-one equivalent of a formal program.

Inclusion requires an investment identifier, sufficient historical CIO-risk observations to form future labels, and sufficient project or spending data to calculate at least the core predictor set. Observations with impossible denominators, contradictory dates, duplicate keys, or unrecoverable outcome labels are excluded according to a preregistered rule set. Fields with substantial but nonfatal missingness are retained only if missingness can be handled transparently and a missingness indicator is justified.

The Federal IT Dashboard is the primary public source. Its investment pages expose downloadable project and variance source data and historical CIO-rating data. Dashboard documentation defines cost variance as a function of planned versus projected or actual costs and schedule variance as planned versus projected or actual completion timing relative to project duration. It categorizes absolute variance magnitudes below 10% as low, 10% to below 30% as

medium, and at least 30% as high. CIO ratings range from 1 to 5, with higher values reflecting better assessments of the investment's ability to meet current-year goals.
The primary dependent variable is:

$$Y_{i,t,h} = \{1, if\ investment\ i\ records\ a\ CIO\ rating\ of\ 1\ or\ 2\ within\ horizon\ h\ 0, otherwise$$

The principal horizon is 90 days. Sensitivity analyses use 30-, 60-, and 180-day horizons. A secondary outcome equals one when the CIO rating declines by at least one point during the horizon. A second family of secondary outcomes identifies subsequent high cost or schedule variance. The primary endpoint is fixed before model selection to prevent outcome shopping.

Table 1: Proposed variables and operational definitions

Variable	Level	Operationalization	Source	Expected predictive role
CIO risk rating	Investment-period	Historical rating from 1 to 5	Historic CIO Rating feed	Basis of primary future label
Cost variance	Project-period	Signed Dashboard cost variance	Project and Variance data	H2
Absolute cost variance	Project-period	$\|cost\ variance\|$	Derived	Magnitude of departure
Schedule variance	Project-period	Signed Dashboard schedule variance	Project and Variance data	H1
Absolute schedule variance	Project-period	$\|schedule\ variance\|$	Derived	Magnitude of departure
Variance slope	Investment-period	Time slope over previous 3 or 6 observations	Derived	H3
Variance volatility	Investment-period	Rolling standard deviation	Derived	H3
Active-project count	Investment-period	Number of contemporaneously active projects	Derived	H4
Project-cost concentration	Investment-period	Largest-project cost divided by total observed project cost	Derived	H4
Total planned project cost	Investment-period	Sum across included projects	Derived	Scale control
CIO-rating momentum	Investment-period	Difference or slope across prior ratings only	Derived	Persistence and deterioration
Spending growth	Investment-period	Lagged spending change where feed coverage permits	Spending feed	Resource-pressure indicator
Agency	Investment	Categorical fixed/context effect	Investment metadata	Context control
Missingness indicators	Observation	Flags for selected unavailable fields	Derived	Data-quality control

The study explicitly preserves both the sign and absolute magnitude of cost and schedule variance. This is important because the Dashboard formula can distinguish favorable from unfavorable deviations, while magnitude captures general instability. A model using only absolute values would lose directional information.



Fig 2: Study design and model-validation architecture

Data extraction proceeds through a versioned workflow. Raw source files are stored unchanged with retrieval date, source identifier, and checksum. A staging layer standardizes investment identifiers, project identifiers, dates, and numeric fields. A validation layer checks date ranges, duplicate records, impossible project durations, nonfinite ratios, and extreme values. A feature layer creates only information available at or before the prediction timestamp. A modeling layer receives a frozen analytic table.
Missing-value treatment is determined after profiling. Numeric predictors with modest missingness can be imputed from training-period distributions, with imputation parameters learned only from the training data. Variables with structural absence receive separate category or missingness indicators where theoretically appropriate. Features with extreme missingness are excluded from the core model and may be explored in sensitivity analysis. No value from validation or test periods is used to estimate training imputations.
Extreme-value handling is rule-based rather than discretionary. The real-data pilot demonstrates why. One official project record reported a cost-variance value of -88,000%. That observation is retained in the raw supplementary table but flagged as an extreme reported value and excluded from pilot distribution plots using a preregistered magnitude screen. This does not imply that the source is "wrong." It means the value requires provenance review before it can influence a fitted model. Historical GAO reviews have also identified data-accuracy issues in Federal IT Dashboard reporting, which makes transparent data-quality auditing methodologically necessary [27, 28].

Feature engineering is conducted at time t. For variable x, rolling means, slopes, changes, and volatility are calculated strictly from $\{x_{t-k},\ldots,x_t\}$. No forward-filled value may cross the prediction date. Program-level project features aggregate projects that were known to exist at time t. This rule prevents a common but serious form of target leakage in which later project outcomes become predictors of earlier risk. The temporal split is fixed before model tuning. A recommended configuration is training on the earliest usable observations through December 31, 2021, validation on calendar year 2022, and independent testing on January 1, 2023 through December 31, 2024, subject to actual feed coverage. If early coverage is too sparse, a rolling-origin design is used while retaining 2024 as the final untouched evaluation period. Observations from the same date never appear simultaneously on both sides of a temporal boundary.

Table 2: Four model families are compared.

Method	Purpose	Strength	Principal risk	Role in study
Logistic regression	Statistical baseline	Interpretability and probability output	Limited nonlinear interaction capacity	Mandatory benchmark
Decision tree	Transparent nonlinear benchmark	Easily communicated threshold logic	Instability and overfitting	Secondary benchmark
Random forest	Ensemble nonlinear prediction	Captures interactions and nonlinearities, robust aggregation [17]	Less transparent without explanation tools	Candidate final model
Gradient boosting	Sequential nonlinear ensemble	High predictive flexibility [18]	Tuning sensitivity and possible overfitting	Candidate final model

For logistic regression, regularization strength is tuned over a predetermined grid. For random forests, the primary tuning parameters are number of trees, maximum depth, minimum leaf size, and feature subsampling. For gradient boosting, the primary parameters are learning rate, number of boosting iterations, leaf complexity, and L2 regularization. Tuning uses only the training and validation periods. The final test set remains untouched until the model and operating threshold are frozen.

Class imbalance is handled primarily through class weighting and operating-threshold selection. Synthetic oversampling, if examined, is restricted to a sensitivity analysis and performed only inside training folds. It is never applied before temporal splitting. This is important because rare adverse-risk states are precisely the observations where leakage or duplicated information can most seriously exaggerate apparent performance.

Model performance is assessed using recall, precision, specificity, F1-score, ROC-AUC, PR-AUC, Brier score, calibration plots, and confusion matrices. Because potentially adverse CIO outcomes may be uncommon, PR-AUC and precision-recall curves receive special emphasis [19]. The final operating threshold is selected on the validation period according to a prespecified cost rule that gives greater weight to false negatives than false positives. No threshold is optimized on the test set.

The proposed study adds an operational metric, warning lead time. For each correctly predicted adverse episode, lead time equals the number of days between the first model alert above the locked threshold and the first subsequent adverse CIO state. Median lead time and its distribution quantify whether statistical discrimination translates into managerial time to act.

Interpretability is evaluated at two levels. Logistic-regression coefficients provide global direction and magnitude on the log-odds scale. For ensemble models, permutation importance provides a model-level importance check and SHAP values explain individual high-risk predictions [20]. Explanations are constrained to observed features and accompanied by a warning that feature attribution does not establish causal responsibility.

Robustness analyses include alternative prediction horizons, alternative high-risk definitions, signed versus absolute variance specifications, agency-stratified performance, removal of extreme observations, complete-case versus imputed models, and rolling-origin temporal validation. A model is considered operationally credible only if discrimination, calibration, and major feature patterns remain reasonably stable.

Reliability and validity are addressed through reproducible source acquisition, versioned data transformations, fixed feature definitions, transparent exclusion rules, independent temporal testing, and complete retention of the raw-to-analytic transformation code. Construct validity receives special scrutiny because a major federal IT investment is an empirical proxy for a large-scale technology program rather than a universally accepted definition. Harrison and colleagues' 2024 review reinforces the need to make the scale construct explicit rather than treating it as obvious [10].

Results

The present results are deliberately divided into actual feasibility results and illustrative future reporting structure.

The actual pilot was assembled from completed project rows exposed on selected official Federal IT Dashboard investment pages. The sample contains project records from multiple federal investments, including technology, infrastructure, modernization, application, and program-increment work. Only records with planned end dates no later than 2024 were used in the pilot analytical file.

The raw convenience pilot contains 51 completed project records. One record with an absolute reported cost variance above 1,000% was retained for provenance but excluded from descriptive modeling plots, resulting in 50 analyzed observations. These 50 projects represent $550.58 million in total planned cost. Mean planned project cost was $11.01 million, the median was $3.645 million, and the standard deviation was $21.79 million. The distribution is therefore strongly right-skewed.

Table 3: Actual real-data pilot results

Measure	Pilot result
Raw completed project records	51
Quality-screened records used in plots	50
Total planned cost	$550.58 million
Mean planned cost	$11.012 million
Median planned cost	$3.645 million
Mean absolute cost variance	4.921%
Median absolute cost variance	0.000%
Projects with absolute cost variance ≥10%	8 of 50, 16%
Projects with absolute cost variance ≥30%	3 of 50, 6%
Mean absolute schedule variance	4.276%
Median absolute schedule variance	0.000%
Projects with absolute schedule variance ≥10%	6 of 50, 12%
Projects with absolute schedule variance ≥30%	2 of 50, 4%
Projects with both absolute variances ≥10%	1 of 50, 2%
Spearman r_s, planned cost vs. absolute cost variance	0.410
Spearman r_s, absolute schedule vs. absolute cost variance	-0.209

The pilot correlations are exploratory and should not be interpreted as population effects or causal relationships. The positive rank association between planned project cost and absolute cost variance in this small convenience sample is useful only as a demonstration that program-scale variables deserve examination in the complete panel. Likewise, the weak negative pilot association between absolute schedule and cost variance does not establish that schedule and cost risk move in opposite directions. The sample was assembled for feasibility, not inference.

Using the Dashboard's published variance categories, 37 of the 50 quality-screened observations fell below 10% absolute variance on both cost and schedule. Five had medium cost variance with low schedule variance. Four had medium schedule variance with low cost variance. Two had high cost and low schedule variance. One had high schedule and low cost variance, and one had both high cost and high schedule variance. This distribution confirms the need to examine class balance before fitting any rare-risk classifier.

The predictive model-performance table below is synthetic and illustrative only. Its values are included solely to show the reporting architecture requested in the brief. They are not findings, must not appear in a submitted article as empirical results, and must be replaced after the complete historical panel is constructed.

Table 4: Illustrative model-performance reporting template, synthetic values only

Model	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC	Brier score
Logistic regression	0.80	0.58	0.68	0.63	0.82	0.55	0.15
Decision tree	0.76	0.52	0.64	0.57	0.74	0.46	0.18
Random forest	0.84	0.66	0.74	0.70	0.87	0.64	0.13
Gradient boosting	0.86	0.69	0.81	0.75	0.90	0.70	0.11

The F1 values in the illustrative table are mathematically consistent, within rounding, with the stated precision and recall. The ranking is intentionally plausible rather than claimed. A real final model should not be selected solely because it has the largest ROC-AUC. For a risk-warning application, recall, PR-AUC, calibration, lead time, interpretability, and the decision cost of false negatives are equally important [19].

Discussion

The feasibility results establish three points without overstating what the pilot can prove. First, the public data contain actual variation in cost and schedule performance rather than only uniformly successful records. Second, the raw data can contain extreme reported values that require explicit provenance checks and sensitivity analysis. Third, the high proportion of low-variance observations in this convenience sample indicates that adverse-event prediction may face imbalance, reinforcing the case for precision-recall analysis, threshold design, and calibration rather than headline accuracy alone.

The broader literature suggests why an integrated model may be useful. Barki, Rivard, and Talbot established a measurable multidimensional view of software risk [1]. Schmidt and colleagues demonstrated substantial practitioner consensus about important software-project risk factors [2]. Wallace, Keil, and Rai showed that risk dimensions form a sociotechnical sequence rather than behaving as isolated factors [3]. Kutsch and Hall demonstrated that the existence of formal project-risk processes does not guarantee effective practice [4]. The predictive framework extends this tradition by asking whether observable manifestations of those risk processes leave statistical traces early enough to support intervention [1-4].

At the program and portfolio level, the proposed framework also extends prior research rather than competing with it. Teller and Kock found that risk transparency and coping capacity are linked to portfolio success and that formalization, monitoring, and integration matter [7]. Martinsuo and colleagues showed that uncertainty at portfolio level is interpreted and acted upon through managerial framing [8]. Predictive analytics can be conceptualized as an additional mechanism for increasing transparency and directing scarce managerial attention. It does not eliminate framing or judgment. Rather, it changes the information environment within which those managerial processes operate [7, 8].

The federal context is especially useful because risk and performance can be observed repeatedly. Mishra, Das, and Murray's longitudinal study of federal technology projects demonstrated that complexity, contracting, execution risk, and process maturity can be investigated across repeated project periods [6]. GAO's major-IT assessments independently show that substantial program cost and schedule changes occur in practice [5]. The proposed paper differs by making future adverse-state prediction and warning lead time the central estimand rather than estimating contemporaneous or explanatory relationships [5, 6].

The study's theoretical contribution will be strongest if the final data show that deteriorating combinations of operational indicators predict subsequent escalation after controlling for prior CIO risk state. Such a finding would support the proposition that program risk is partly observable as a trajectory, not just as a static score. This would extend sociotechnical risk thinking from relationships among risk categories [3] toward a temporal information-processing view in which risk manifestations evolve and become more or less observable over time.

The methodological contribution is the architecture rather than any particular algorithm. Random forests and gradient boosting are established methods [17, 18]. Their application alone is not novel. Novelty instead comes from the longitudinal label design, explicit prediction horizon, temporally ordered validation, multi-level program feature construction, leakage controls, calibration, warning-lead-time metric, and decision-oriented explainability.

Managerially, the system should operate as a decision-support mechanism rather than an automated escalation engine. A probability estimate identifies attention priority. A driver explanation identifies which observations contributed to the estimate. A manager then assesses whether those indicators reflect a known temporary condition, a data-quality anomaly, or a true emerging threat. This human-review requirement is consistent with broader responsible-AI principles emphasizing governance, measurement, management, and trustworthy use [26].

Table 5: The proposed action architecture is:

Predicted condition	Illustrative governance response	Human review requirement
Low calibrated risk	Continue routine monitoring	Verify data freshness
Moderate risk or rising trajectory	Review top model drivers and open risks	Program analyst or PM review
High risk	Structured risk review, schedule/cost root-cause analysis, intervention options	Program manager and PMO
Critical risk with persistent signal	Executive escalation, resource or scope decision, independent assurance where appropriate	Named accountable executive

The probability boundaries should not be fixed at arbitrary 25%, 50%, or 75% cutoffs before empirical calibration. Thresholds should be selected from the validation data based on false-negative costs, alert capacity, calibration, and governance tolerance.

The final dashboard should show the current calibrated risk probability, change since the prior reporting period, prediction horizon, uncertainty or calibration information, top contributing risk drivers, schedule and cost trends, previous alerts, and the disposition of prior interventions. This design makes model use auditable. It also enables a later research question concerning whether managers act on alerts and whether interventions change outcomes.

Conclusion

Large technology programs generate enough operational information that risk management need not be limited to retrospective reporting or periodic qualitative assessment. Existing research establishes that technology risk is multidimensional, sociotechnical, affected by complexity, embedded in portfolios and programs, and dependent on managerial interpretation [1-10, 21-24]. Predictive methods provide additional tools for detecting nonlinear combinations and temporal deterioration [15-20]. The research opportunity is to integrate these bodies of knowledge without confusing prediction with causation or automation with governance.

The proposed framework therefore moves through a disciplined sequence: historical data acquisition, provenance control, time-aware feature engineering, future-state labeling, chronological model development, independent testing, calibration, explainability, management review, intervention, and continuous monitoring. Its principal scientific test is not whether a sophisticated algorithm can fit historical project data. It is whether information available before a risk escalation can predict that escalation reliably enough, early enough, and transparently enough to improve managerial attention and decision support.

The real-data pilot establishes the feasibility of extracting meaningful cost and schedule variation from public federal IT records while also exposing data-quality challenges that a credible empirical article must address. The full contribution can be claimed only after the historical CIO-rating panel is assembled and the locked predictive protocol is executed. At that point, the strongest possible conclusion would be evidence that technology-program risk can be treated as a continuously monitored, partially observable dynamic state and that data analytics can complement established risk-governance processes with empirically validated early-warning intelligence.

Limitations

The public federal setting limits external validity to private-sector technology environments. A major IT investment is a proxy for a large-scale program rather than a universal formal definition. Historical CIO ratings incorporate managerial judgment and therefore are not purely objective labels. Public feeds may omit important scope, workforce, vendor, defect, cybersecurity, stakeholder, and dependency variables. Reporting practices may differ by agency and year. Historical Dashboard accuracy concerns require explicit audit controls [27, 28]. Predictive performance may drift over time as reporting systems and technology-delivery practices change. None of the models establishes causal effects. Finally, the pilot sample is a convenience feasibility sample and cannot support population-level inference.

Future Work

Future research should validate the framework across private-sector industries and geographic settings, integrate risk-register and issue-log text through natural-language features, represent dependencies using network or graph models, investigate model drift and recalibration, evaluate causal effects of interventions, and conduct controlled studies of human-AI risk decision making. Another extension should compare predictive risk management with prescriptive optimization, while preserving executive accountability for high-consequence decisions. Research should also examine whether predictive signals create undesirable behaviors, such as excessive escalation or gaming of project metrics.

References

1. Barki H, Rivard S, Talbot J. Toward an assessment of software development risk. Journal of Management Information Systems. 1993;10(2):203-225. doi:10.1080/07421222.1993.11518006
2. Schmidt R, Lyytinen K, Keil M, Cule P. Identifying software project risks: An international Delphi study. Journal of Management Information Systems. 2001;17(4):5-36. doi:10.1080/07421222.2001.11045662
3. Wallace L, Keil M, Rai A. How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model. Decision Sciences. 2004;35(2):289-321. doi:10.1111/j.00117315.2004.02059.x
4. Kutsch E, Hall M. Intervening conditions on the management of project risk: Dealing with uncertainty in information technology projects. International Journal of Project Management. 2005;23(8):591-599. doi:10.1016/j.ijproman.2005.06.009
5. U.S. Government Accountability Office, IT Systems Annual Assessment: DOD Needs to Improve Performance Reporting and Development Planning, GAO-23-106117, reissued with revisions Aug. 22, 2024.
6. Mishra A, Das SR, Murray JJ. Risk, process maturity, and project performance: An empirical analysis of US federal government technology projects. Production and Operations Management. 2016;25(2):210-232. doi:10.1111/poms.12513
7. Teller J, Kock A. An empirical investigation on how portfolio risk management influences project portfolio success. International Journal of Project Management. 2013;31(6):817-829. doi:10.1016/j.ijproman.2012.11.012
8. Martinsuo M, Korhonen T, Laine T. Identifying, framing and managing uncertainties in project portfolios. International Journal of Project Management. 2014;32(5):732-746. doi:10.1016/j.ijproman.2014.01.014
9. Shi Q, Zhou Y, Xiao C, Chen R, Zuo J. Delivery risk analysis within the context of program management using fuzzy logic and DEA: A China case study. International Journal of Project Management. 2014;32(2):341-349. doi:10.1016/j.ijproman.2013.05.002
10. Harrison JW, Lewis MA, Roehrich JK, Davies A. Scale in project management: A review and research agenda. Project Management Journal. 2024;55(6):708-722. doi:10.1177/87569728241256611
11. Keil M, Tiwana A, Bush A. Reconciling user and project manager perceptions of IT project risk: A Delphi study. Information Systems Journal. 2002;12(2):103-119. doi:10.1046/j.1365-2575.2002.00121.x
12. Calisir F, Gumussoy CA. Determinants of budget overruns on IT projects. Technovation. 2005;25(6):631-636. doi:10.1016/j.technovation.2003.10.011
13. Zwikael O, Pathak RD, Singh G, Ahmed S. The moderating effect of risk on the relationship between planning and success. International Journal of Project Management. 2014;32(3):435-441. doi:10.1016/j.ijproman.2013.07.002
14. Uzzafer M. A contingency estimation model for software projects. International Journal of Project Management. 2013;31(7):981-993. doi:10.1016/j.ijproman.2012.12.002
15. Alatawi MN, Alyahyan S, Hussain S, Alshammari A, Aldaeej AA, Alali IK, *et al*. A data-driven artificial neural network approach to software project risk assessment. IET Software. 2023;2023:4324783. doi:10.1049/2023/4324783
16. Moussa A, Ezzeldin M, El-Dakhakhni W. Predicting and managing risk interactions and systemic risks in infrastructure projects using machine learning. Automation in Construction. 2024;168:105836. doi:10.1016/j.autcon.2024.105836
17. Breiman L. Random forests. Machine Learning. 2001;45:5-32. doi:10.1023/A:1010933404324
18. Friedman JH. Greedy function approximation: A gradient boosting machine. The Annals of Statistics. 2001;29(5):1189-1232. doi:10.1214/aos/1013203451
19. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE. 2015;10(3):e0118432. doi:10.1371/journal.pone.0118432
20. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems. 2017;30:4765-4774.
21. Ahern T, Leavy B, Byrne PJ. Complex project management as complex problem solving: A distributed knowledge management perspective. International Journal of Project Management. 2014;32(8):1371-1381. doi:10.1016/j.ijproman.2013.06.007
22. Bjorvatn T, Wald A. Project complexity and team-level absorptive capacity as drivers of project management performance. International Journal of Project Management. 2018;36(6):876-888. doi:10.1016/j.ijproman.2018.05.003
23. Lee JS, Keil M, Shalev E. Seeing the trees or the forest? The effect of IT project managers' mental construal on IT project risk management activities. Information Systems Research. 2019;30(3):1051-1072. doi:10.1287/isre.2019.0853
24. Lehtinen TOA, Mäntylä MV, Vanhanen J, Itkonen J, Lassenius C. Perceived causes of software project failures: An analysis of their relationships. Information and Software Technology. 2014;56(6):623-643. doi:10.1016/j.infsof.2014.01.015

25. International Organization for Standardization, ISO 31000:2018 Risk Management: Guidelines, 2nd ed., 2018.
26. E. Tabassi, Artificial Intelligence Risk Management Framework, AI RMF 1.0, NIST AI 100-1, National Institute of Standards and Technology, 2023, doi: 10.6028/NIST.AI.100-1.
27. U.S. Government Accountability Office, Information Technology: OMB Has Made Improvements to Its Dashboard, but Further Work Is Needed by Agencies and OMB to Ensure Data Accuracy, GAO-11-262, 2011.
28. U.S. Government Accountability Office, Information Technology: OMB's Dashboard Has Increased Transparency and Oversight, but Improvements Needed, GAO-10-701, 2010.

How to Cite This Article
Sodamola AP, Omoifo-Irefo MO, Obioha CB. Predictive risk management in large-scale technology programs: a data analytics framework for early risk detection and decision support. International Journal of Artificial Intelligence Engineering and Transformation. 2025;6(1):35-44. doi:10.54660/IJAIET.2025.6.1.35-44.

Creative Commons (CC) License
This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms