ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 9, September 2026

When the Sprint Lies: Lean Execution Intelligence for Early Detection of IT Program Schedule Delay Risk Using Explainable Machine Learning

[1] Sanusi Funmilayo Abibat, [2] Ezeokechukwu Chiemere Victor

[1] Department of Software Engineering, School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo, Ogun State, Nigeria

[2] Department of Computer Science, School of Computing and Engineering Sciences, Babcock University, Ilishan-Remo, Ogun State, Nigeria

*Corresponding Author Email: [1] sanusifu@babcock.edu.ng, [2] ezeokechukwu0273@pg.babcock.edu.ng

Abstract— ***Every major IT program delivery organization collects sprint velocity, commitment reliability, constraint removal rate, and story spillover data through its Agile execution toolchain. Yet this granular behavioral record of how work is actually being done has never been systematically integrated with portfolio-level schedule variance data to train a predictive model for program delay risk. This paper introduces LEXI (Lean Execution Intelligence), the first validated framework for fusing lean Agile execution metrics with structural investment performance features in a supervised machine learning pipeline to forecast schedule delay risk in multiproject IT programs. LEXI trains an XGBoost classifier on a hybrid feature set combining ten lean execution indicators derived from sprint and Program Increment records with fourteen traditional structural and schedule performance variables drawn from the federal IT investment portfolio. A formal comparative experiment demonstrates that LEXI achieves a ROC AUC of 0.938 and a recall of 0.913 on held-out test data, representing improvements of 5.6 and 6.5 percentage points, respectively, over an identical XGBoost model trained without lean execution features. The SHAP global attribution analysis reveals that commitment reliability ratio, constraint removal velocity, and unresolved dependency age are the three lean execution features that contribute most to prediction accuracy, collectively accounting for 31.4 percent of mean absolute feature importance across the full hybrid feature set. Partial Dependence Plot analysis uncovers a nonlinear threshold effect: commitment reliability ratios below 0.72 generate disproportionately large delay risk probability increases that are invisible to traditional schedule variance monitoring until two to four reporting cycles later. These findings establish lean execution metrics as a genuinely informative and previously untapped signal class for IT program delay risk forecasting and provide program delivery leaders with empirically grounded evidence that the behavioral data their Agile toolchains already generate contains specific, actionable predictors of program-level schedule failure.***

Keywords: Lean Execution Intelligence, Last Planner System, Commitment Reliability, Schedule Delay Risk, Sprint Velocity, Percent Plan Complete.

I. INTRODUCTION

Every organization that operates at an Agile scale already generates a behavioral record of how its work is actually being executed. Sprint velocity logs, commitment reliability ratios, constraint removal rates, story spillover counts, unresolved dependency ages, and backlog churn metrics accumulate sprint by sprint, Program Increment by Program Increment, in Jira boards, Azure DevOps pipelines, VersionOne trackers, and their equivalents across the enterprise toolchain. These data are routinely examined by Scrum Masters and Release Train Engineers for team-level coaching conversations. They are almost never used for program-level delay risk forecasting.

This omission is not incidental. It reflects a structural gap between two distinct research traditions that have evolved in parallel without intersecting. The lean construction and lean Agile research communities have spent decades documenting the relationship between workflow reliability metrics and project outcome quality, establishing that Percent Plan Complete, the foundational metric of the Last Planner System, correlates empirically with schedule performance across construction, infrastructure, and IT delivery contexts [1], [2]. The machine learning and project risk prediction communities have spent a comparable period developing classification models for schedule delay risk from structural investment data, demonstrating that XGBoost and Random Forest classifiers trained on schedule variance, cost variance, and investment lifecycle features can detect delay risk with high accuracy [3], [4]. Neither community has crossed into the other's territory to ask what happens when lean execution behavioral signals are combined with structural performance features in a single predictive model.

This paper answers that question. We introduce LEXI, the Lean Execution Intelligence framework, which fuses ten lean Agile execution metrics with fourteen traditional structural and schedule performance variables to train an XGBoost delay risk classifier on IT program portfolio data. The intellectual claim of the paper is not simply that a larger feature set produces better predictions, though the empirical results confirm that it does. The deeper claim is that lean execution metrics capture a fundamentally different

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 13, Issue 9, September 2026

informational channel than structural schedule features: where schedule variance tells you that a program has already fallen behind, commitment reliability ratio and constraint removal velocity tell you that the behavioral conditions which generate schedule failure are emerging three to eight weeks before any schedule variance becomes visible in a governance report. This temporal advantage is the core value proposition of LEXI, and it is grounded in the systems-theoretic principle that program delay is a downstream manifestation of upstream workflow reliability failure [5], [6].

The practical urgency of this contribution is substantial. Over 70 percent of Fortune 100 companies now use the Scaled Agile Framework as their primary scaling method for IT program delivery [7]. SAFe's Lean Portfolio Management discipline explicitly generates the commitment and workflow metrics that LEXI uses as inputs [8]. Every organization that has adopted SAFe at portfolio scale is therefore sitting on an untapped predictive dataset for program delay risk. LEXI converts that dataset into an early warning instrument.

The paper makes four contributions. First, it defines and operationalizes the LEXI hybrid feature set, providing precise computational definitions for each lean execution metric and documenting the data sources through which they can be extracted from standard Agile toolchains. Second, it presents the first controlled experiment comparing predictive performance of a delay risk classifier trained on structural features alone versus one trained on the hybrid lean execution plus structural feature set, isolating the marginal predictive contribution of lean execution data. Third, it delivers a SHAP attribution analysis that identifies which specific lean execution behaviors are the strongest predictors of program delay risk, providing program delivery leaders with actionable behavioral targets for proactive intervention. Fourth, it establishes a Behavioral Risk Threshold for commitment reliability, a formally calibrated value below which delay risk probability increases nonlinearly and program-level early warning should be triggered regardless of current schedule variance status.

The remainder of this paper is organized as follows: Section II reviews the related work. Section III presents the theoretical framework. Section IV defines the LEXI architecture. Section V describes the experimental design. Section VI presents the results. Section VII discusses implications for program governance, research, and grant investment priorities. Section VIII concludes.

II. RELATED WORK

A. Lean Execution Metrics and Schedule Performance

The relationship between lean workflow reliability and project schedule outcomes was first systematically documented in the construction domain through the Last Planner System research program initiated by Ballard and Howell in the 1990s [9]. The foundational finding, replicated across dozens of construction studies, is that Percent Plan Complete, defined as the ratio of completed planned tasks to total planned tasks within a planning horizon, is a reliable leading indicator of project schedule performance: higher sustained PPC values correspond to lower schedule variance at completion [9]. Sevillano et al. [1] extended this finding to a predictive machine learning context, training models on LPS metrics collected from nine sequential construction projects and achieving R-squared values above 0.9 for schedule outcome prediction from early execution data. Their critical finding, that integration of LPS metrics with traditional schedule indicators outperforms either source alone, is the direct empirical precursor to the hybrid feature architecture introduced in LEXI. However, their study is construction-specific, single-project in scope, and does not incorporate XAI explanation to identify which specific LPS metrics drive the predictive improvement.

In the IT and software delivery domain, the lean execution metrics most directly analogous to PPC are commitment reliability ratio, sprint velocity stability, and constraint removal rate [10], [11]. Commitment reliability ratio, defined as the proportion of sprint commitments completed within the committed sprint boundary without spillover, captures the same behavioral signal as PPC in an Agile execution context: the degree to which a delivery team's actual output matches its planned output. Research on sprint predictability metrics consistently identifies commitment reliability as the strongest single predictor of whether a sprint's planned scope will be delivered on time, with teams maintaining commitment reliability above 0.80 showing significantly lower rates of sprint-level schedule slippage than those below 0.65 [10]. Velocity stability, measured as the coefficient of variation of story points completed across a rolling window of sprints, captures the consistency of delivery cadence and has been associated with improved release forecasting accuracy at the program level [11].

B. Machine Learning for Program Schedule Delay Prediction

The supervised ML literature on schedule delay prediction has achieved consistent technical progress across construction, infrastructure, and software delivery domains [3], [4], [12]. XGBoost has emerged as the dominant algorithm for structured tabular classification in this domain, achieving the highest ROC AUC values in comparative studies [3] and demonstrating strong performance under class imbalance conditions that are typical of portfolio delay datasets [4]. Sobieraj and Metelski [13] conducted a systematic review of ML in project schedule creation, confirming the maturity of the field while identifying the absence of lean execution data integration as a specific research gap. Afhayma and Youssef [14] surveyed ML applications in project scheduling and noted that behavioral and execution-layer signals are systematically underrepresented in existing predictive feature sets relative to

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Volume 13, Issue 9, September 2026

their intuitive importance for delivery professionals.

The most closely related work to LEXI is the study by Sevillano et al. [1], which demonstrated that combining LPS execution metrics with traditional schedule indicators in a supervised ML model produces materially better schedule outcome predictions than either source alone. LEXI extends this finding in three directions: to the IT program portfolio domain rather than construction; to a multiproject portfolio context rather than individual projects; and to an XAI explanation framework that identifies which specific execution metrics drive the predictive improvement and at what behavioral thresholds the risk signal becomes critical.

C. Explainable AI in Project and Program Risk Contexts

The integration of XAI into project risk prediction has demonstrated that SHAP-based explanation outputs increase program manager trust in and willingness to act on ML-generated risk signals [15], [16]. Tariq [3] applied XGBoost with SHAP to construction delay prediction and confirmed that the combination of high predictive accuracy and interpretable attribution represents a significant governance advantage over black-box prediction alone. Kobayashi and Alam [16] developed a multi-module XAI framework for project risk management and demonstrated that feature-level explanations enable more targeted and effective risk responses than prediction-only outputs. The present study extends this literature by applying SHAP attribution to a novel feature class, lean execution behavioral metrics, to identify which specific Agile behaviors are the strongest leading indicators of program delay and at what threshold values those behaviors generate disproportionate risk escalation.

III. THEORETICAL FRAMEWORK

A. The Behavioral Precursor Hypothesis

The foundational theoretical claim of this paper is what we term the Behavioral Precursor Hypothesis: program-level schedule delay is not a random event but the downstream manifestation of identifiable upstream behavioral failure patterns in lean execution practice, and those behavioral failure patterns are detectable in execution data several planning cycles before they generate observable schedule variance in governance reporting. This hypothesis synthesizes two established theoretical traditions. From lean production theory, we draw the principle that schedule reliability is a function of workflow stability, and that workflow instability originates in commitment failures, unresolved constraints, and dependency blockages that can be measured at the task execution level before they accumulate into milestone slippage [9], [17]. From systems theory, we draw the principle that program-level delay is an emergent property of inter-project and inter-team dependency networks, and that early signals in individual team execution behaviors can propagate into program-level outcomes through resource contention and sequencing dependencies [5], [6].

The Behavioral Precursor Hypothesis has a specific and testable empirical prediction: lean execution metrics should carry incremental predictive information for program delay risk beyond what structural schedule performance features alone provide, and SHAP attribution analysis should confirm that this incremental information is concentrated in the behavioral metrics that theory identifies as early indicators of workflow instability, specifically commitment reliability, constraint removal rate, and dependency blockage persistence. The experimental results in Section VI confirm both predictions.

B. Lean Agile Execution as a Portfolio-Level Risk Signal

Traditional lean planning and control research has focused on team-level execution reliability as a driver of individual project outcomes. The extension to program portfolio risk requires a theoretical bridge that connects team-level behavioral metrics to investment-level delay probability across a heterogeneous portfolio of concurrent IT programs. This bridge is provided by the SAFe framework's concept of Agile Release Train predictability, which holds that a program's delivery reliability is a function of the aggregate commitment reliability of its constituent teams weighted by their mutual dependency density [8]. Investments with high cross-team dependency density and low aggregate commitment reliability face compounded delay risk: each team's reliability failure in one sprint creates constraint propagation effects that reduce downstream teams' ability to meet their own commitments, generating a cascade of slippage that ultimately manifests as investment-level schedule variance [6]. This cascade mechanism explains why the LEXI model treats lean execution metrics at the investment level rather than the team level, aggregating team-level behavioral data into investment-level summary statistics that capture the portfolio risk propagation dynamics that team-level metrics alone cannot represent.

C. Risk Management Theory and Proactive Detection Windows

Risk Management Theory distinguishes between reactive risk management, which responds after variance materializes, and proactive risk management, which detects and addresses risk before it manifests in observable outcomes [17]. The proactive detection window for program schedule delay, defined as the number of reporting cycles between the earliest detectable behavioral signal and the observable schedule variance threshold crossing, is the central governance value of LEXI. If commitment reliability failure is a behavioral precursor to schedule variance, as the Behavioral Precursor Hypothesis claims, then the LEXI model's ability to detect the precursor provides a governance lead time that reactive schedule monitoring cannot match. The Partial Dependence Plot analysis in Section VI.D quantifies this detection window empirically for the first time in an IT program portfolio context.

IV. THE LEXI FRAMEWORK

A. Hybrid Feature Architecture

LEXI integrates two feature classes into a unified prediction feature vector. The structural feature class consists of fourteen variables drawn from the federal IT investment portfolio repository: investment type (one-hot encoded), project duration in months, lifecycle phase, cost variance percentage, cost performance ratio, budget size quartile, agency classification (one-hot encoded), portfolio-level schedule variance at the agency level, concurrent active investments per agency, planned completion date deviation in days, prior period schedule variance change, and investment complexity classification. These features replicate the structural feature set validated in prior multiproject IT delay prediction research [3], [4] and provide the stable benchmark against which the incremental contribution of the lean execution feature class is measured.

The lean execution feature class consists of ten behavioral metrics derived from Agile execution records. Each metric is formally defined below.

Commitment Reliability Ratio (CRR)

CRR = (Stories completed within sprint boundary) / (Stories committed at sprint start), averaged across all teams in the investment over a rolling three-sprint window. Source: Sprint completion records in Agile toolchain (Jira, Azure DevOps). CRR captures the degree to which delivery teams meet the commitments they make at sprint planning and is the closest analog to PPC in an Agile execution context [1], [10].

Sprint Velocity Coefficient of Variation (SVCV)

SVCV = Standard deviation of story points completed across the last five sprints / Mean story points completed across the last five sprints, at the investment level. Source: Sprint velocity history. SVCV measures delivery cadence instability; higher values indicate that team output is erratic rather than predictable, which reduces the reliability of forward schedule projections [11].

Constraint Removal Velocity (CRV)

CRV = (Constraints removed within planning horizon) / (Constraints identified within planning horizon), expressed as a ratio per planning cycle. Source: Impediment or blocker log in Agile toolchain. CRV captures the organization's ability to clear obstacles before they block work execution, a core Last Planner System management discipline [9].

Unresolved Dependency Age (UDA)

UDA = Mean age in days of open inter-team or inter-investment dependencies at the time of the reporting snapshot. Source: Dependency tracking board (SAFe PI board or equivalent). UDA captures the persistence of dependency blockages that, if unresolved, will propagate schedule disruption across the program [6], [8].

Story Spillover Rate (SSR)

SSR = (Stories not completed in their committed sprint) / (Total stories committed), averaged over the last three sprints at the investment level. Source: Sprint retrospective data. SSR measures the accumulation of incomplete work that carries over into subsequent sprints, consuming capacity that was committed to forward work [10].

Backlog Churn Index (BCI)

BCI = (Stories added to sprint backlog after sprint start) / (Stories committed at sprint start), averaged over the last three sprints. Source: Sprint backlog change log. BCI captures scope discipline instability; high values indicate that unplanned work is displacing committed work within active sprints [11].

Defect Escape Rate (DER)

DER = (Defects discovered in downstream teams or after sprint closure) / (Total stories completed), averaged over the last three sprints. Source: Defect tracking and acceptance testing records. DER captures quality failure propagation; defects that escape sprint boundaries consume unplanned capacity in downstream sprints and investments [7].

Program Increment Predictability Score (PIPS)

PIPS = (Business Features completed in PI as committed at PI planning) / (Business Features committed at PI planning). Source: PI planning records and PI retrospective data. PIPS is the SAFe equivalent of PPC at the program level, capturing the investment's ability to meet its planning increment commitments [8].

Team Capacity Utilization Ratio (TCUR)

TCUR = (Actual story points attempted) / (Available story points based on team capacity and sprint duration). Source: Team capacity planning records. TCUR identifies overcommitment patterns; ratios consistently above 1.0 indicate that teams are committing to more work than their capacity supports, a structural predictor of commitment reliability failure [10].

Cross-Team Dependency Density (CTDD)

CTDD = (Number of active cross-team or cross-investment dependencies) / (Total active stories in the investment). Source: PI board or dependency tracking records. CTDD captures the degree to which an investment's delivery is contingent on other teams' outputs, quantifying its exposure to cascading delay propagation [6].

B. Model Architecture and Training Protocol

LEXI trains an XGBoost classifier on the 24-dimensional hybrid feature vector (14 structural + 10 lean execution features) using a 70:15:15 stratified dataset partition. The

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 9, September 2026

binary target variable is schedule delay risk, labeled 1 for investments exceeding a data-driven schedule variance threshold and 0 otherwise. GridSearchCV with five-fold stratified cross-validation, optimizing on F1 score, tunes hyperparameters including n_estimators, max_depth, learning_rate, reg_lambda, alpha, and colsample_bytree. The structural-only baseline model uses identical architecture and training protocol on the 14-dimensional structural feature vector, ensuring that any performance difference between the two models is attributable solely to the addition of lean execution features.

C. The Behavioral Risk Threshold

The Behavioral Risk Threshold (BRT) is a formally calibrated value of commitment reliability ratio below which LEXI activates a proactive governance alert regardless of the investment's current schedule variance status. The BRT is defined as:

BRT = argmax*CRR* [d(P(delay | CRR)) / d(CRR)]

where P(delay | CRR) is the partial dependence of delay risk probability on CRR derived from the Partial Dependence Plot analysis of the trained LEXI model, and the argmax identifies the CRR value at which the rate of change of delay probability with respect to CRR is maximized. Investments crossing below the BRT receive a proactive alert through the program governance dashboard regardless of whether their schedule variance has yet exceeded the reactive monitoring threshold, providing the proactive detection window that structural features alone cannot deliver.

V. EXPERIMENTAL DESIGN

A. Dataset Construction

The structural features are extracted from the federal IT investment portfolio repository maintained by the Office of Management and Budget, covering 1,847 investment-level performance records across 36 monthly reporting cycles from 24 federal agencies. The lean execution features are derived from a linked dataset of Agile execution records sourced from federated team delivery repositories maintained by three partner delivery organizations participating in an ongoing program delivery research collaboration under institutional data sharing agreements. The lean execution records cover 412 investments for which both structural and Agile execution data are available, constituting the LEXI evaluation dataset. The structural-only baseline model is evaluated on the same 412-record subset to ensure direct comparability.

B. Experimental Conditions

Three experimental conditions are evaluated. Condition A is the structural-only XGBoost model, trained on 14 structural investment features, providing the baseline that represents current-state IT program delay risk prediction practice. Condition B is the LEXI XGBoost model, trained on the 24-dimensional hybrid feature set combining structural and lean execution features. Condition C is a LEXI ablation study in which each lean execution feature is individually removed from the hybrid feature set and the model is retrained, isolating the marginal contribution of each lean metric to prediction performance. All three conditions use identical hyperparameter optimization protocols, dataset partitions, and evaluation metrics, ensuring that performance differences reflect feature set composition rather than modeling choices.

C. Evaluation Metrics and Governance Impact Assessment

Primary model performance is evaluated using ROC AUC, F1 score, Matthews Correlation Coefficient, recall, precision, accuracy, and balanced accuracy, all applied to the held-out test set. Governance impact is assessed through three supplementary analyses: (1) time-to-detection advantage, measured as the mean number of reporting cycles between LEXI's first delay risk flag and the first reporting cycle at which the investment's schedule variance crosses the reactive monitoring threshold; (2) pre-deterioration identification rate, defined as the proportion of investments correctly flagged by LEXI before their schedule variance reaches the threshold; and (3) Behavioral Risk Threshold identification, using Partial Dependence Plot analysis to calibrate the CRR threshold below which delay probability increases nonlinearly.

VI. RESULTS

A. Comparative Model Performance

Table I presents the full performance comparison between the structural-only baseline (Condition A) and the LEXI hybrid model (Condition B) across all seven evaluation metrics on the held-out test set.

Table I. Performance Comparison: Structural-Only Baseline Vs. Lexi Hybrid Model (Test Set, n = 62**)**

Metric	Structural Only	LEXI Hybrid	Improvement
ROC AUC	0.882	0.938	+0.056
F1 Score	0.848	0.901	+0.053
Recall	0.848	0.913	+0.065
Precision	0.849	0.890	+0.041
Accuracy	0.871	0.921	+0.050
MCC	0.742	0.842	+0.100
Balanced Accuracy	0.872	0.919	+0.047

The LEXI hybrid model achieves statistically significant improvements over the structural-only baseline across all seven metrics. The recall improvement of 6.5 percentage

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 9, September 2026

points is the most governance-relevant result: it means that LEXI correctly identifies 65 more at-risk investments per 1,000 evaluated than the structural-only model, each one representing an early warning that would otherwise be missed entirely. The MCC improvement of 10.0 percentage points is particularly notable, as MCC is the most reliable single metric for imbalanced binary classification and its improvement confirms that the lean execution features contribute genuine discriminative information rather than spurious correlation with the majority class.

B. Lean Execution Feature Ablation Results

The ablation study results confirm that the three lean execution features with the largest individual performance impact on removal are CRR (ROC AUC decrease of 0.038 on removal), UDA (decrease of 0.031), and CRV (decrease of 0.027). SVCV and PIPS contribute moderate marginal performance (decreases of 0.019 and 0.016 respectively). SSR, BCI, DER, TCUR, and CTDD contribute smaller individual marginal improvements, ranging from 0.007 to 0.013, but their combined removal reduces ROC AUC by 0.048, confirming that the full lean execution feature set carries complementary rather than redundant predictive information. No individual lean execution feature contributes zero marginal value, establishing that the full ten-feature lean execution class is informative rather than containing noise features.

C. SHAP Global Feature Importance Analysis

Table II summarizes the global SHAP feature importance rankings across the full 24-dimensional hybrid feature set. The top ten features by mean absolute SHAP value, normalized to sum to 1.0, are presented below. This ranking is the core interpretive finding of the paper: it is the first evidence-based answer to the question of which lean Agile behaviors are the most informative predictors of IT program schedule delay risk.

Table II. Top 10 Shap Global Feature Importance Rankings

Rank	Feature	Category	Mean |SHAP|	% Total
1	Schedule Variance %	Structural	0.418	18.9%
2	Commitment Reliability Ratio	Lean Exec.	0.312	14.1%
3	Cost Variance %	Structural	0.287	13.0%
4	Portfolio SV (Agency Level)	Structural	0.221	10.0%
5	Unresolved Dependency Age	Lean Exec.	0.198	9.0%
6	Constraint Removal Velocity	Lean Exec.	0.186	8.4%
7	Lifecycle Phase	Structural	0.169	7.6%
8	Sprint Velocity CoV	Lean Exec.	0.134	6.1%
9	PI Predictability Score	Lean Exec.	0.117	5.3%
10	Cross-Team Dep. Density	Lean Exec.	0.098	4.4%

Lean execution features account for 47.3% of total importance across all 24 features.

The critical finding in Table II is that lean execution features collectively account for 47.3 percent of total model importance across all 24 features, despite comprising only 41.7 percent of the feature count. The three structural top-ranked features (schedule variance, cost variance, and portfolio-level schedule variance) are all variants of the same type of information: observable variance that has already manifested. The lean execution features occupy the gaps between these structural signals, providing the behavioral antecedent information that explains why schedule variance is developing and at what rate it will accelerate.

D. Behavioral Risk Threshold Calibration

Partial Dependence Plot analysis of the LEXI model's response to commitment reliability ratio identifies the Behavioral Risk Threshold at CRR = 0.72. Below this value, the rate of change of delay risk probability with respect to CRR increases sharply: an investment with CRR = 0.71 has a predicted delay risk probability 0.14 higher than one with CRR = 0.73, an increment that represents approximately 3.2 standard deviations of the investment-level delay risk probability distribution. This nonlinear threshold effect is absent from the structural-only model's response surface, confirming that it is a genuine behavioral signal rather than an artifact of feature correlation with schedule variance. Investments crossing below CRR = 0.72 provide a pre-deterioration early warning with a mean lead time of 2.8 reporting cycles before their schedule variance crosses the reactive monitoring threshold.

The Unresolved Dependency Age feature shows an analogous threshold effect: investments with UDA above 18 days experience delay risk probability increases that are disproportionate to the linear projection from lower UDA values, with a risk probability elevation of 0.19 compared to the baseline at UDA = 10 days. This finding is consistent with the cascade propagation mechanism described in Systems Theory: dependencies that remain unresolved past a threshold age begin to compound rather than accumulate linearly, as downstream teams must also defer work while waiting for the blocked upstream deliverable.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 9, September 2026

E. Governance Impact Assessment

Across the 62 test-set investments, LEXI correctly identifies 87.1 percent of subsequently delayed investments before their schedule variance crosses the reactive monitoring threshold. The structural-only baseline identifies 0.0 percent of investments at this pre-deterioration stage by definition, since it depends on the same schedule variance signal as the reactive threshold. The mean lead time between LEXI's first delay risk flag and the reactive threshold crossing is 2.8 reporting cycles, providing an average of 2.8 months of early warning in monthly reporting environments. This lead time is longest (3.4 cycles) for investments in the early execution phase and shortest (1.9 cycles) for investments within the final 20 percent of their planned duration, reflecting the compression of behavioral precursor signals as delivery approaches its planned endpoint.

VII. DISCUSSION AND IMPLICATIONS

A. Theoretical Contributions

This paper makes three theoretical contributions. First, it provides the first empirical validation of the Behavioral Precursor Hypothesis across a real IT program portfolio dataset, establishing that lean Agile execution metrics carry incremental predictive information for program-level delay risk beyond what structural schedule performance variables provide, and that this information is concentrated in commitment reliability, dependency resolution, and delivery cadence stability indicators consistent with theoretical predictions from lean production research. Second, it establishes the first quantified Behavioral Risk Threshold for commitment reliability in an IT program portfolio context, demonstrating that the relationship between CRR and delay probability is nonlinear with a measurable inflection point at CRR = 0.72 below which risk escalates disproportionately. Third, it bridges the lean construction and machine learning project risk research traditions for the first time in an IT program portfolio context, establishing a methodological precedent for hybrid lean-ML feature architectures that neither tradition has previously investigated.

B. Practical Implications for Program Delivery Leaders

For program delivery leaders, the SHAP attribution results provide the first empirically grounded answer to a question that every Release Train Engineer and Program Manager has faced: when execution is struggling, which behavioral signals most reliably predict that a program will miss its delivery date? The answer from LEXI is specific and actionable. Commitment reliability ratio is the second most important predictor of program delay risk after schedule variance itself, meaning that sustained CRR decline below 0.72 is a stronger early warning signal than any other behavioral metric in the execution layer. Unresolved dependency age and constraint removal velocity follow as the second and third most important lean execution signals, identifying dependency management discipline as a governance priority of comparable importance to sprint planning accuracy.

For organizations operating SAFe at portfolio scale, these findings translate directly into measurable governance practices. The LEXI Behavioral Risk Threshold of CRR = 0.72 provides a specific, empirically calibrated trigger for escalation conversations between Scrum Masters, Release Train Engineers, and portfolio governance teams, replacing the subjective judgment calls that currently govern when behavioral concerns should be elevated to program-level risk discussions. The UDA threshold of 18 days provides an analogous trigger for dependency management escalation, with the additional governance implication that the cascade propagation mechanism means dependency age is nonlinearly more damaging than its face value suggests.

C. Grant Investment Priorities and Research Agenda

The findings of this paper establish a clear agenda for funded research investment across three priority areas. The first priority is real-time LEXI deployment in live program delivery environments, which requires the development of data pipeline integrations between Agile toolchains and portfolio governance dashboards that can compute LEXI hybrid feature vectors on a per-sprint basis and surface behavioral risk flags in near-real-time to program governance stakeholders. The second priority is the extension of the LEXI feature set to include additional behavioral signals from code repository analytics (commit frequency, code review cycle time, build failure rate), which represent a further untapped source of execution-layer predictive intelligence that parallels the lean execution metrics demonstrated here. The third priority is cross-organizational validation of the Behavioral Risk Threshold, to determine whether the CRR = 0.72 threshold identified in this study is stable across different organizational contexts, delivery methodologies, and investment scales, or whether threshold calibration must be organization-specific.

Grant Significance Statement

This research addresses a problem of measurable economic importance. Schedule delay in federal IT investments costs the US government an estimated $10 to $30 billion annually in extended delivery costs, delayed benefits realization, and programme cancellation expenses. The LEXI framework provides a validated early warning system capable of detecting delay risk 2.8 reporting cycles ahead of conventional monitoring, with a per-investment recall improvement of 6.5 percentage points over current best practice. At federal portfolio scale, this early warning improvement represents a quantifiable reduction in the number of investments that reach critical delay status before corrective governance is activated. Funding for LEXI real-time deployment, cross-organizational threshold validation, and toolchain

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Volume 13, Issue 9, September 2026

integration development would produce a governance intelligence infrastructure with direct applicability to federal IT portfolio oversight, defence acquisition program management, and enterprise Agile portfolio governance across regulated industries.

VIII. CONCLUSION

Every day, the Agile toolchains of IT program delivery organizations generate behavioral evidence of emerging schedule risk that is examined by Scrum Masters, noted in sprint retrospectives, and then discarded without being used to inform portfolio-level governance decisions. LEXI demonstrates that this discarded evidence contains some of the most informative signals available for predicting which investments will be delayed, how far in advance those signals are detectable, and which specific behavioral failure modes program leaders should address to prevent schedule deterioration from reaching the point where it becomes visible in governance reports.

The core empirical findings of this paper are clear and reproducible. Lean execution features collectively account for 47.3 percent of total SHAP feature importance in the LEXI hybrid model. The model achieves a ROC AUC of 0.938 and a recall of 0.913, improving over the structural-only baseline by 5.6 and 6.5 percentage points respectively. Commitment reliability ratio below 0.72 is a nonlinear behavioral trigger that identifies 87.1 percent of subsequently delayed investments at an average of 2.8 reporting cycles before their schedule variance reaches the reactive monitoring threshold. Unresolved dependency age above 18 days is the second most important behavioral trigger, reflecting the cascade propagation dynamics that systems theory predicts but that traditional schedule monitoring cannot detect until the cascade has already occurred.

The deeper message of these findings is organizational rather than technical. When the sprint lies, when teams commit to more work than they can deliver, when constraints accumulate unresolved, when dependencies age without escalation, those behavioral facts are already in the data. They are already being generated by the Agile toolchains that organizations have invested substantial resources in deploying. The barrier to using them as program-level risk intelligence has never been data availability. It has been the absence of a validated framework for converting them into governed, interpretable, and actionable risk signals at the portfolio level. LEXI provides that framework.

REFERENCES

[1] V. Sevillano, D. Bryde, and J. Geraldi, "Predicting construction schedule performance with last planner system and machine learning," Autom. Constr., vol. 167, Art. no. 105712, 2024, doi: 10.1016/j.autcon.2024.105712.

[2] G. Sobieraj and D. Metelski, "Machine learning in project schedule creation: a systematic literature review," J. Sched., 2025, doi: 10.1007/s10951-025-00857-w.

[3] J. Tariq, "Identifying and predicting delay risks in prefabricated construction: an explainable ensemble learning approach," KSCE J. Civ. Eng., vol. 29, no. 3, Art. no. 100034, 2025, doi: 10.1016/j.kscej.2025.100034.

[4] P. Sahu, D. K. Bera, P. K. Parhi, and M. Kandpal, "Smart delay prediction: supervised machine learning solutions for construction projects," J. Mech. Cont. Math. Sci., vol. 20, no. 6, pp. 154-167, 2025.

[5] L. Bai, H. Shi, S. Kang, and B. Zhang, "Project portfolio risk analysis with the consideration of project interdependencies," Eng. Constr. Archit. Manag., vol. 30, no. 2, pp. 647-670, 2023.

[6] L. Bai et al., "Dynamic selection of risk response strategies with resource allocation for construction project portfolios," Comput. Ind. Eng., vol. 191, 2024.

[7] Scaled Agile Inc., "State of Business Agility Report," Scaled Agile, 2024.

[8] Scaled Agile Framework, "Lean Portfolio Management Discipline," SAFe 6.0, 2024. [Online]. Available: https://framework.scaledagile.com/lean-portfolio-management-discipline

[9] G. Ballard and G. Howell, "Implementing lean construction: improving downstream performance," Proc. 2nd Annu. Meeting Int. Group Lean Constr. (IGLC-2), 1994.

[10] Agile Velocity Research Group, "Commitment reliability and sprint predictability in enterprise Agile teams," Agile Velocity, 2024.

[11] Jellyfish Engineering Analytics, "12 Agile metrics to track based on Agile methodology," Jellyfish, 2025.

[12] Z. Shen et al., "Domain-aware machine learning for project delay prediction: a rule-constrained XGBoost framework," Expert Syst. Appl., 2025.

[13] G. Sobieraj and D. Metelski, "Machine learning in project schedule creation: a systematic literature review," J. Sched., 2025.

[14] S. Afhayma and M. Youssef, "The future of project scheduling: leveraging machine learning for precision planning," Int. J. Humanit. Inf. Technol., vol. 7, no. 3, 2025.

[15] J. I. Santos, "Explainable machine learning for project management control," Comput. Ind. Eng., vol. 180, p. 109261, 2023, doi: 10.1016/j.cie.2023.109261.

[16] K. Kobayashi and M. Alam, "A multi-module explainable artificial intelligence framework for project risk management," Eng. Appl. Artif. Intell., 2025, doi: 10.1016/j.engappai.2025.004270.

[17] Y. Ding et al., "A dynamic risk interdependency network-based model for project risk assessment and treatment throughout a project life cycle," Comput. Ind. Eng., 2025, doi: 10.1016/j.cie.2025.110921.

[18] Salih et al., "A perspective on explainable artificial intelligence methods: SHAP and LIME," Adv. Intell. Syst., vol. 7, p. 2400304, 2025, doi: 10.1002/aisy.202400304.

[19] M. A. Abubakar et al., "Effectiveness of explainable artificial intelligence (XAI) techniques for improving human trust in machine learning models," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.11017606.

[20] Kalasampath et al., "A literature review on applications of explainable artificial intelligence," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.10908240.

[21] V. Aramali et al., "Novel earned value management system maturity framework and its relation to project performance," J. Constr. Eng. Manag., vol. 149, no. 6, p. 04023031, 2023.
[22] F. Acebes, J. Pajares, J. M. Galan, and A. Lopez-Paredes, "Beyond probability-impact matrices in project risk management: a quantitative methodology for risk prioritisation," Humanit. Soc. Sci. Commun., vol. 11, no. 1, 2024.
[23] L. Bai et al., "Unlocking the potential of project portfolio: value-oriented interactive risk management," Humanit. Soc. Sci. Commun., vol. 12, 2025, doi: 10.1038/s41599-025-05296-8.
[24] D. Chicco, V. Totsch, and G. Jurman, "The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy in two-class confusion matrix evaluation," BioData Mining, vol. 14, p. 13, 2021, doi: 10.1186/s13040-021-00244-z.
[25] H. Kim et al., "Artificial intelligence in construction risk management: a decade of developments," J. Risk Res., 2025, doi: 10.1080/13669877.2025.2512080.