Applied Artificial Intelligence Research

INNOVIAIR

From Risk Prediction to Decision Quality: AI-Driven Agile Sprint Planning Assessment under Capacity Uncertainty

Yushuai Cong[a,*]

[a]School of Business, Belarusian State University, Minsk 220030, Belarus

ARTICLEINFO

Keywords:
Agile
Sprint
Forecasting
Calibration
Optimization

ABSTRACT

Sprint planning requires a choice between committing tasks and reserving capacity for work added during an iteration. We examined how added-task forecasts affect this decision using 238 Sprints from four public Jira projects: 129 for training, 37 for calibration, and 72 for later evaluation. For 38 eligible Sprints, we compared reservation policies through a common mixed-integer task-selection model under simulated workload and capacity shocks. Forecasting accuracy did not favor the more complex models: mean absolute error (MAE) was 5.853 for the seven-feature random forest (RF7), 4.556 for Lag-1, and 5.097 for the post hoc six-feature refit (RF6). At 50% simulated risk tolerance, the random-forest strategy had higher counterfactual proxy loss, calculated from closing-report task attributes, than fixed 20% reservation (7.079 versus 5.974; paired difference +1.105, exploratory 95% interval [0.553, 1.684]). Its simulated replanning incidence was lower in point estimate (39.47% versus 43.86%; difference −4.39 percentage points, 95% interval [−8.77, 0.00]), but the interval reached zero. The paired intervals for replanning and loss versus calibrated Lag-1 also crossed zero. The 38-Sprint comparisons and 54 sensitivity settings indicate a trade-off between simulated replanning incidence and schedulable proxy task value. Because the task pools and nominal capacities were derived from Sprint closing reports, these findings describe retrospective scenario comparisons rather than observed delivery outcomes.

1.Introduction

Sprint planning involves two related decisions: the work a team can commit to and the capacity it should reserve for tasks arriving during the iteration. A small reserve can lead to changes in the plan; a large reserve can defer higher-priority work. An added-task forecast is therefore useful only when its implications for both plan stability and scheduled task value are evaluated.

Predict-then-optimize research separates forecasting error from downstream optimization loss[1]. Agile task-selection methods address task priorities and capacity constraints[2], and cross-Sprint planning models consider replanning explicitly[3]. Random forests provide a forecasting benchmark[4]. Decision-focused learning also distinguishes prediction quality from decision quality[5]. We compare learning-based and historical reservation rules using the same candidate tasks and optimization constraints. The evaluation measures simulated replanning incidence alongside the proxy task value displaced by reservation.

Historical observations and counterfactual planning outcomes require separate interpretation. Public Sprint summaries record added-task counts at Sprint close, but do not provide complete planning-date task lists, contemporaneous estimates, or actual team capacity. We construct the candidate pool from closing reports and vary workload and capacity shocks explicitly. The resulting comparisons describe the specified offline scenarios, rather than effects observed after deployment.

We address three research questions. RQ1: How does a random forest's forecast of added-task counts compare with a simple previous-Sprint rule? RQ2: Under the same candidate tasks, capacity, and simulated shocks, how do reservation rules differ in replanning risk and counterfactual proxy loss? RQ3: What can Sprint-level paired comparisons, parameter sensitivity analysis, and decision-time information audits tell us about these differences? Forecasting and downstream decisions are evaluated separately; retrospective task selection is not treated as a reconstruction of actual Sprint-start commitments.

* Corresponding author.
E-mail address: yushuai.cong01@gmail.com.
https://doi.org/10.65455/18dg5h61
Received 20 October 2026; Received in revised form 13 November 2026; Accepted 18 November 2026; Available 24 November 2026
© The Author(s) 2026. This article is published with open access in Applied Artificial Intelligence Research under a Creative Commons Attribution 4.0 International License (CC BY 4.0). https://creativecommons.org/licenses/by/4.0/

2.Related research

2.1.Agile task selection and replanning

Özçelikkan et al.[2] proposed a multi-objective Scrum planning method involving priorities, capacity utilization, and story grouping. Golfarelli et al.[3] studied planning and smooth replanning across Sprints. We use an established binary task-selection formulation as a common decision platform. This isolates the effects of reservation estimates from changes in the optimizer itself.

2.2.Prediction, calibration, and downstream decision making

Random forests are established nonlinear prediction methods[4]. Elmachtoub and Grigas[1] and Mandi et al.[5] distinguish prediction error from the loss incurred by subsequent optimization decisions. We follow that distinction by comparing random forests with historical rules under the same mixed-integer task-selection model, measuring simulated replanning incidence and the proxy task value displaced by reserved capacity.

2.3.Time series evaluation and model selection bias

Chronological training, calibration, and evaluation splits help prevent future information from entering historical forecasts. Their empirical bounds may still vary with project characteristics and data distributions. We therefore interpret residual-based bounds as conditional estimates for the current sample, not as coverage guarantees for other projects. Gibbs and Candès[6] examine distribution shift. Cerqueira et al.[7] study time dependence in forecasting evaluation, and Cawley and Talbot[8] identify selection-induced optimism. Because the 38 decision instances were examined repeatedly during method development, the reported intervals are exploratory and require independent validation.

2.4.Implications of prediction – optimization research for agile sprint planning

This paper uses three kinds of research ideas for reference: selecting Sprint tasks under priority and capacity constraints, evaluating prediction errors and subsequent decision losses, and verifying the information available at the prediction or planning time point. To compare the capacity reservation rules, we keep the candidate task pool, optimizer, and constraints fixed, jointly report the incidence of simulated replanning and the proxy value of the displaced tasks, and separately audit the timing of task attributes.

3.Data and methods

3.1.Data sources, chronological order design, and boundaries of evidence

The research data comes from the public AgileScrumSprintVelocityDataSet[9], including four Jira projects, Aurora, Mesos, Spring XD and UserGrid. After cleaning, 238 Sprints were retained and divided into 129 training samples, 37 calibration samples and 72 later-evaluation samples in chronological order within each project. Among them, 38 later Sprints have the records needed to build retrospective task-selection instances: Aurora 2 (5.3%), Mesos 6 (15.8%), Spring XD 19 (50.0%), and UserGrid 11 (28.9%). Spring XD accounts for half of the decision samples. Another 22 calibration Sprints were used for preliminary analysis. Table 1 describes the purpose and scope of evidence of various types of data, and Table 3 lists the audit of project-level task records. These samples are from the same batch of public data and are not a new independent external test set.

The forecast target is the number of added tasks reported in each Sprint's closing summary. Reliable historical records are not available for the complete planning-date task list, actual team capacity, or story-point estimates at the moment of task addition. We therefore use observed closing-report attributes together with explicitly defined decision scenarios. The auxiliary Itemlet data comprise 1,598 static groups across 68 projects and serve only as task-structure and feasibility evidence; the dataset is described in a repository record[10] and a companion data descriptor[11]. These historical forecasts, static checks, and simulated decisions are distinct forms of evidence.

Table 1. Dataset structure and evidence boundaries

Data subset	Sample size	Permitted use	Not established
Cleaned Sprint records	238 Sprints, 4 projects	Historical prediction of added-task counts	Generalization across organizations
Training / calibration / later evaluation	129 / 37 / 72	Chronological model training and evaluation	Distribution-free coverage guarantees
Eligible evaluation Sprints	38 Sprints, 4 projects	Counterfactual task selection	Actual Sprint-start commitments
Itemlet auxiliary static groups	1,598 groups, 68 projects	Task-structure and static-feasibility checks	Complete within-Sprint event history

Note: The four-project Jira summaries were cleaned from source[9]. The auxiliary Itemlet data are documented in its dataset record[10] and companion publication[11]. Chronological training and evaluation within the same four projects is not independent external validation.

3.2.Audit of information availability and time validity

The prediction model and task-selection model depend on information from different time points. The added-task forecast uses only historical Sprint summary records completed before the start of the target Sprint; the forecast target is the number of new tasks in the target Sprint closing report. The original field issueKeysAddedDuringSprint records the number of added tasks at Sprint level, not the specific task identities and entry times in the optimization candidate pool. Because the archive lacks a complete membership snapshot at the planning time, the task-selection model uses the tasks and attributes in the target Sprint closing report to build a candidate pool. Although initialStoryPoint records the initial story-point estimate of a task, it cannot establish that the task already belonged to the task pool when the target Sprint starts. Therefore, closing-report story points

and priorities should not be regarded as verified planning-time inputs.

The final records of 38 decision Sprints contain 1,056 task – Sprint associations, involving 806 distinct IssueKeys; 913 of these associations have positive story points and sortable priorities. In the same project, matching IssueKeys against reports completed before the target Sprint began identifies 247 associations, of which 227 belong to the 913 usable associations; the remaining 809 associations had no match in earlier reports. Neither of the two types of records is sufficient to identify membership at target-Sprint start: an earlier occurrence does not establish continued membership in the current Sprint, and failure to find a historical match does not mean that the task was added during the current Sprint. The original exported data and replication scripts do not provide a task-by-task, timestamped Sprint entry and exit history. Under the current strict verification criterion, all 1,056 associations lack confirmed Sprint-start membership status. Table 3 records it as 100% "undeterminable" instead of 100% "mid-Sprint addition". In addition, the added-task count in the closing report and the filtered MILP candidate records use different denominators, which cannot be directly divided to calculate the actual added-task share in the candidate pool.

Table 2. Audit of decision-time information availability

Data element	Temporal evidence	Permissible interpretation
Sprint boundaries and prior summaries	Dated records for completed Sprints	May construct historical predictor features
Added-task count for the target Sprint	End-of-Sprint summary	Prediction outcome only
Target-Sprint task list / story points / priority	Terminal task report; effective times unverified	Retrospective candidate pool for counterfactual analysis
Issue keys observed in earlier Sprint reports	247 of 1,056 issue–Sprint associations	Evidence of prior appearance, not confirmed initial commitment
Actual team capacity and arriving workload	Not reliably observed	Represented using explicit simulation assumptions

Note: Based on the time-traceability audit of 38 closing reports. Evidence of an earlier appearance is not proof of Sprint-start commitment.

Table 3. Project-level traceability and summary-added-task-share audit

Project	Evaluation Sprints	Closing task–Sprint associations	Traceable in prior reports	Mean summary-level added-task share	Undeterminable ratio
Aurora	2	29	2/29 (6.90%)	60.00%	100% (not certified)
Mesos	6	75	45/75 (60.00%)	28.29%	100% (not certified)
Spring XD	19	677	131/677 (19.35%)	27.45%	100% (not certified)
UserGrid	11	275	69/275 (25.09%)	38.54%	100% (not certified)
Total / overview	38	1,056	247/1,056 (23.39%)	32.51% (Sprint mean)	100% (not certified)

Note: historical traceability means that the same IssueKey has appeared in an earlier-completed Sprint report in the same project. Among the 1,056 closing-report task–Sprint associations of 38 decision Sprints, 247 had an earlier-report match and 809 did not. The summary-level added-task share is calculated for each Sprint as AddedDuringSprint / IssueCountReported and then averaged across Sprints. This indicator comes from the closing summary and is not the timestamp-confirmed proportion of new members in the MILP candidate pool. "100% undeterminable" means that the audited associations lack verifiable target-Sprint-start membership records, which does not mean that all tasks were added during the Sprint. A match to an earlier report cannot independently prove that the task belongs to the initial task pool of the current Sprint.

3.3.Comparison of two sets of historical experimental protocols and a unified approach

The early study used two protocols. Protocol C combined random-forest forecasts and empirical upper bounds with a greedy task-selection rule under moderate simulated stress. Protocol E compared seven reserve ratios derived from previous-Sprint counts and historical residuals and used mixed-integer optimization under three simulated shocks. Because their selection rules and scenario definitions differ, we report them separately. Cross-policy interpretation rests primarily on the later unified model.

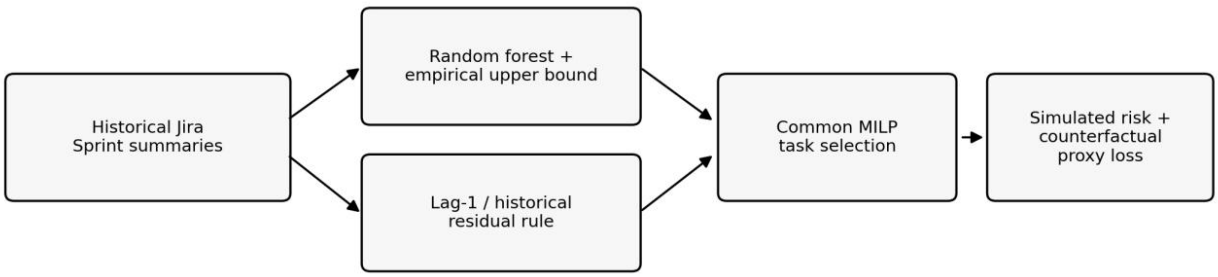

Fig 1. Unified prediction-to-decision evaluation workflow. Forecasting rules vary while the downstream MILP task-selection model and simulation settings remain fixed.

Note: Conceptual workflow illustrating the experimental design, not observed organizational performance.

In the unified comparison, we held fixed the candidate tasks, nominal capacity, simulated pressure, priority-value mapping, and the SciPy HiGHS 0 – 1 mixed-integer optimizer, allowing only the risk estimation and reservation policy to

change. The prediction model was first trained on 129 sprints, then calibrated on 37 sprints, and its prediction performance was evaluated on 72 later sprints; 38 of these sprints had the conditions for constructing optimization instances. These 38 samples were examined repeatedly during the study. We therefore treat the unified policy comparisons as exploratory; independent validation would require new data. Figure 1 illustrates the research flow from new task prediction to task selection.

3.4.Added-task forecasting and empirical risk bounds

Let A_t be the number of new tasks reported in the final report of the t-th sprint, and X_t be the historical features provided by the sprints already completed before the start, including the number of tasks in the previous sprint and short-term rolling statistics. The prediction given by the random forest is $f_{RF}(X_t)$; the simple historical baseline directly uses the number of new tasks $A_{(t-1)}$ from the previous sprint. We then calculate the empirical upper quantile $q_{(1-\alpha)}$ based on the residual e_j of "actual value minus predicted value" in the calibration sample, to form an empirical upper demand estimate for capacity reservation.

$$U_t = \lceil \max(0, f(X_t) + q_{1-\alpha}) \rceil \quad (1)$$

We evaluate nominal 80% and 90% empirical upper bounds using their observed coverage and mean upper-bound size in the later sample. These percentages are calibration targets; coverage under different projects or task distributions must be evaluated separately. We report point-forecast accuracy using mean absolute error (MAE), where lower values indicate smaller average absolute errors.

3.5.Capacity reservation and binary task selection model

For candidate task i, take the story-point estimate s_i and priority p_i in its final report, and take $v_i = 6 - p_i$ as a simplified proxy for task value. The nominal capacity is calculated based on the total number of story points in the candidate pool at the end of the same period, $C = 0.72\ \Sigma_i\ s_i$; this value is a model setting, not the actual capacity of the team. The reserve ratio is selected from R = {0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30}. Given the ratio r, each strategy uses the same 0–1 mixed-integer task-selection model:

$$\max_x \sum_i v_i x_i \quad (2)$$

$$\sum_i s_i x_i \le C(1 - r), \quad x_i \in \{0,1\} \quad (3)$$

Let $V(r)$ be the maximum task priority-proxy value that can be scheduled under the reserve ratio r using closing-report task attributes, and define $L(r) = V(0) - V(r)$, hereinafter referred to as "counterfactual proxy loss". It represents the schedulable proxy value reduced by reserved capacity for a specified task pool and simulated capacity; it cannot be interpreted as the observed revenue loss or the actual opportunity cost on the planning day. For low, medium, and high simulated shocks, this paper tests whether the sum of the original selected workload and the simulated new workload exceeds the remaining capacity after the shock, and calculates the incidence of simulated replanning. Each reservation strategy runs according to the calibrated rules; if none of the candidate ratios can meet the risk threshold τ, the prespecified minimum-predicted-risk fallback is adopted, and failure to meet the risk target is reported.

The closing-report task pool provides consistent inputs for policy comparison, but it may also cause a timing mismatch. Tasks entering a Sprint after planning may be included in the simulated initial selection; if a task priority or story point changes during execution, the closing value may differ from the planning-time value. The nominal capacity is calculated from the total number of story points in the task pool at the end of the same period, so changes to the candidate pool will also change the capacity benchmark. If the priority of a task is raised during Sprint execution, its final proxy score may increase. However, the existing records can neither quantify the overall deviation nor establish that most tasks with raised priorities were added mid-Sprint. Each strategy uses common inputs and constraints, and can be compared under the same retrospective setting; if a verified Sprint-start candidate pool is used, the MILP task-selection and policy rankings may still change. This paper interprets these results as a relative comparison under a given counterfactual structure, and does not treat them as observed organizational planning performance.

$$\hat{R}_t(r) = \text{mean}_{e,\beta}\ 1\{\text{plannedLoad}(r) + \text{addedLoad}(e,b) > \text{availableCapacity}(b)\} \quad (4)$$

The comparison covers a historical-distribution rule, calibrated Lag-1, calibrated random forest, a fixed reserve chosen during calibration, and a fixed 20% reference. We first assess whether each policy meets the simulated risk target and then compare counterfactual proxy loss. An additional "omniscient" policy, defined using knowledge of the simulated outcomes, provides a theoretical value benchmark, not an implementable planning-day strategy.

3.6.Sensitivity analysis and inference interpretation

Sensitivity analysis varies nominal capacity coefficients (0.60, 0.72, 0.85), priority-to-value mappings (linear, high-priority amplification, tiered), arrival-workload multipliers (0.75, 1.00, 1.25), and simulated risk tolerance (τ = 0.50 or 0.75). The resulting 3 × 3 × 3 × 2 = 54 settings use value scores [5, 4, 3, 2, 1], [16, 8, 4, 2, 1], or [5, 5, 3, 1, 1] for priority levels 1–5. All settings reuse the same 38 Sprints; they assess sensitivity to assumptions, not 54 independent experiments (Table 4).

The main experiment used 38 Sprints as bootstrap resampling units, kept the three simulated shocks of each Sprint together, and resampled 4,000 times under the established protocol (archived random seed 20261017). The prediction error analysis uses the recovered 72 later-evaluation records to calculate the absolute errors of RF7, Lag-1 and RF6 on the same Sprints. For MAE and its paired differences, we used a project-stratified paired Sprint bootstrap: we resampled 8,000 times, retained the original sample size for each project in every draw, and reported 95% percentile intervals (random seed 20261111). The new protocol replaces the endpoints of the previous MAE intervals, and the two sets of results are not mixed. Empirical upper-bound coverage reports the covered count, evaluated count, and Wilson 95% binomial intervals, and requires cautious interpretation for projects with small samples.

For the subgroup analysis of added-task share, first calculate the "number of added tasks / total reported tasks" from each Sprint summary report, and then rank the 38 decision Sprints by this share and divide them into two groups of 19 Sprints each. The study only reaggregated the results of the existing 114 simulated shock scenarios, and did not retrain the prediction model or change the MILP task-selection model. This grouping is based on the report-level proportion, not the timestamp-verified share of new members in the optimization candidate pool.

Table 4. Sensitivity analysis parameter grid (54 settings; the same 38 Sprints)

Parameter	Levels	Interpretation
Nominal capacity / total candidate story points	0.60 / 0.72 / 0.85	Assumed planning capacity
Priority-to-value mapping	Linear / high-priority emphasis / tiered	Alternative ordinal proxy-value scales
Added-workload multiplier	0.75 / 1.00 / 1.25	Sensitivity to synthetic arrival workload
Simulated risk tolerance	0.50 / 0.75	Target maximum replanning incidence

Note: These 54 combinations vary constructed simulation assumptions and reuse the same 38 Sprints. They are parameter-sensitivity settings, not independent evaluations or observed team capacities.

3.7.External data access and six-feature transferability audit

The original predictor used seven historical features: lag1, mean3, median5, std5, n_history, last_dev, and last_duration. The previous-Sprint developer count, last_dev, could not be reconstructed consistently in the external audit. A six-feature random forest was then refitted to the original 129 training Sprints. It used 250 trees, a minimum leaf size of 4, a maximum feature fraction of 0.8, and random seed 20261014. The 37 calibration and 72 later evaluation Sprints were unchanged. This refit followed inspection of the original results, so RF6 serves as a post hoc feature-sensitivity diagnostic rather than independent transfer evidence.

We also audited historical TAWOS change logs[12], focusing on Sprint membership timestamps, activation dates, and the provenance of historical features. Comparable external evaluation would require independent projects selected after the protocol is frozen, reliable activation times, sufficiently complete membership histories, consistent added-task definitions, and predictors available at the decision time. Events lacking supporting evidence were left missing rather than inferred to enlarge the test set.

For the anonymized Public Jira Dataset[13], released in June 2025, we examined selected metadata, field definitions, and a small archive prefix. The available definitions mention Sprint-related fields but do not establish a complete membership-change timeline. The inspected prefix is not evidence about the contents of the full archive. We therefore treat this dataset as a candidate for later record-level feasibility checks, not as an external prediction test.

4.Results

4.1.Prediction accuracy and empirical upper bound (RQ1)

In the same 72 later-evaluation Sprints, the MAE of the seven-feature random forest RF7 was 5.853, Lag-1 was 4.556, and the post hoc six-feature random forest RF6 was 5.097. 8,000 paired Bootstrap draws (random seed 20261111) were conducted with project stratification, and the 95% interval for RF7 was [4.677, 7.102], Lag-1 was [3.458, 5.792], and RF6 was [4.106, 6.144]. The MAE difference of RF7 minus Lag-1 was +1.298, with an interval of [−0.038, 2.661]; RF6 minus Lag-1 is +0.541, interval [−0.524, 1.607]. The difference intervals of both differences span zero. The private verification archive saved by the author contains all 72 paired predictions and reproducible calculation scripts; The public S1–S4 summary package does not include these record-level data. Public release still requires checking redistribution permissions for the original data. Under the nominal 80% empirical upper bound, RF and Lag-1 both covered 60/72 Sprints (83.3%), and their average upper bounds were 12.50 and 11.39, respectively; the nominal 90% upper bound of RF covered 69/72 Sprints (95.8%), and the average upper bound is 18.49 items. These coverage rates reflect the current evaluation sample.

Table 5. Later-stage forecasting performance and empirical one-sided coverage (72 Sprints)

Method	MAE (tasks)	Observed upper-bound coverage (n/N; Wilson 95% CI)	Mean upper bound (tasks)
Random forest (point forecast)	5.853	—	—
Lag-1 (point forecast)	4.556	—	—
RF 80% empirical upper bound	—	60/72 (83.3%) [73.1%, 90.2%]	12.50
Lag-1 80% empirical upper bound	—	60/72 (83.3%) [73.1%, 90.2%]	11.39
RF 90% empirical upper bound	—	69/72 (95.8%) [88.5%, 98.6%]	18.49

Note: Source and nature: Offline evaluation of the actual sum of new tasks and historical model predictions across 72 late-stage sprints in four projects; MAE and empirical coverage are calculated metrics. For the empirical upper-bound rows, coverage is shown as covered/evaluated (observed percentage), with Wilson 95% binomial confidence intervals. The 80% and 90% labels refer to nominal bound levels.

Table 6. MAE values and exploratory paired-bootstrap intervals over 72 later Sprints.

Model / paired comparison	N	MAE or difference	Exploratory paired-bootstrap 95% interval
Original RF7	72	5.853	[4.677, 7.102]
Lag-1	72	4.556	[3.458, 5.792]
Refitted RF6	72	5.097	[4.106, 6.144]
RF7 − Lag-1	72	+1.298	[−0.038, 2.661]
RF6 − Lag-1	72	+0.541	[−0.524, 1.607]

Note: A project-stratified paired Bootstrap used 72 paired Sprint observations; each draw retained the original project sample sizes. The procedure used 8,000 resamples (random seed 20261111) and reported a 95% interval from the 2.5th to the 97.5th percentile. All methods used the same resampling indices. The record-level predictions and deterministic recalculation scripts are kept in the author's private audit archive, pending redistribution-permission review before public release. The interval only describes the error change in the current sample and does not represent independent external transfer performance.

Table 7. Nominal RF 80% empirical-bound coverage by project

Project	Covered / evaluated	Nominal RF 80% bound coverage	Wilson 95% binomial interval
Aurora	10/12	83.3%	55.2%–95.3%
Mesos	28/29	96.6%	82.8%–99.4%
Spring XD	15/20	75.0%	53.1%–88.8%
UserGrid	7/11	63.6%	35.4%–84.8%
Overall	60/72	83.3%	73.1%–90.2%

Note: Wilson 95% binomial intervals are calculated based on the observed covered counts and evaluated sample sizes for each project. Projects with fewer observations have wider intervals, so the numerator and denominator should be considered together in interpretation

The observed coverage rates of the nominal 80% empirical upper bound of random forest in the four projects were, respectively: Aurora 10/12 (83.3%), Mesos 28/29 (96.6%), Spring XD 15/20 (75.0%), UserGrid 7/11 (63.6%). Table 7 lists the corresponding Wilson 95% binomial intervals. The sample sizes for Aurora and UserGrid are small and their intervals are wide, so these point estimates do not establish a stable coverage difference between projects.

4.2.Earlier protocol comparisons and the omniscient benchmark

In the earlier Protocol C, the fixed 20% reserve averaged 0.342 simulated replanning triggers per Sprint and a counterfactual proxy loss of 6.105. The RF-80% policy yielded 0.447 and 5.632; RF-90% yielded 0.342 and 7.184. Greater reserves can accommodate some shocks while reducing the proxy value schedulable initially. Because Protocol C uses a greedy selection rule, cross-policy conclusions are drawn primarily from the subsequent common-MILP comparison.

Under Protocol E at $\tau = 0.50$, the calibration-selected fixed 20% reserve had simulated replanning incidence of 43.86% and counterfactual proxy loss of 5.974. The historical-residual adaptive rule yielded 45.61% and 6.105. Both met the overall risk target, but the adaptive rule did not have lower loss. At a 25% target, all tested policies exceeded the threshold; even the omniscient reference could not do better than 33.33% incidence. At 43.86% incidence, the omniscient reference could reduce proxy loss from 5.974 to 3.842 (35.7%), indicating both room for improving the planning rule and limits to the attainable risk targets (Table 8).

Table 8. Protocol E comparison: a simulated risk target of 25% was unattainable within the tested reserve actions

Target simulated risk	Fixed reserve selected on calibration data	Fixed strategy: risk / loss	Lag-1 residual-adaptive: risk / loss
25%	30%	33.33% / 9.026	34.21% / 8.605
50%	20%	43.86% / 5.974	45.61% / 6.105
75%	5%	68.42% / 1.553	64.04% / 2.868

Note: Source and nature: Offline planning calculations of the synthetic impact of early protocol E on the historical task structure. Percentages and value losses are not enterprise field-observed performance; reported separately from the unified protocol used in Table 9

4.3.Unified planning comparison: the cost of lower simulated risk (RQ2)

At a simulated risk tolerance of 50%, the fixed 20% reserve yielded an incidence of 43.86% and counterfactual proxy loss of 5.974. The corresponding results were 50.88% and 5.184 for the historical-distribution rule, 41.23% and 6.526 for calibrated Lag-1, and 39.47% and 7.079 for calibrated random forest. The historical-distribution incidence point estimate exceeded the specified target. RF had the lowest incidence point estimate but the highest proxy loss. Because paired risk-difference intervals touch or cross zero, these results do not establish a consistent risk reduction. The point estimates indicate a risk – value trade-off within this sample. Table 9 and Figure 2 report the comparison. RF6 remains a separate forecast diagnostic and was not used to produce these policy results.

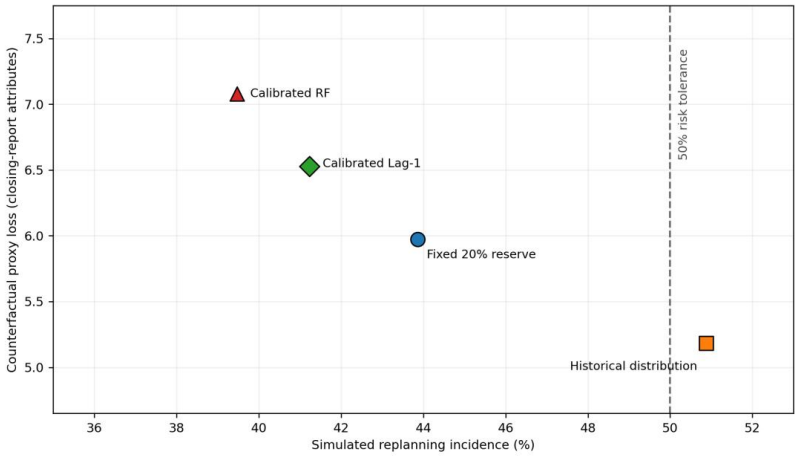

Fig 2. Simulated replanning incidence and counterfactual proxy loss in the unified MILP comparison (lower is better for both measures)

Note: Source: unified main experiment, results/01_unified_policy_summary.csv. Results are computed for 38 Sprints under three synthetic stress scenarios per Sprint (114 observations). The dashed vertical line marks the 50% risk target

Table 9. Common-MILP comparison at a simulated risk tolerance of 50%

Reservation strategy	Mean reserved capacity	Simulated replanning incidence (%)	Counterfactual proxy loss (closing-report attributes)
Calibration-selected fixed 20%	20.00%	43.86	5.974
Historical-distribution rule	16.05%	50.88 (above the 50% target)	5.184
Calibrated Lag-1 rule	18.95%	41.23	6.526
Calibrated RF rule	21.18%	39.47	7.079

Note: Source and nature: Unified MILP master experiment; 38 sprints × 3 synthetic impact scenarios, totaling 114 scenario results. Proxy value is derived based on historical priority and is not actual economic benefit. Source: Unified Comparison Archive results/01_unified_policy_summary.csv

The unified comparison fixed the candidate task pool, nominal capacity and simulated shocks. The fixed 20% reservation strategy had a simulated replanning incidence of 43.86% and a counterfactual proxy loss of 5.974; the random forest strategy had 39.47% and 7.079, respectively. Based on the paired calculation of 38 Sprints, the difference in simulated replanning incidence between the random forest and fixed 20% reservation strategies was −4.39 percentage points, and the exploratory 95% interval was [−8.77, 0.00]; the counterfactual proxy loss difference was +1.105, with an interval of [0.553, 1.684]. The first interval reached zero, while the latter interval was entirely positive. Therefore, in the current sample, the increase in proxy loss is clearer than the decline in risk.

Compared with calibrated Lag-1, the random forest risk difference was −1.75 percentage points [−7.02, 3.51], and the proxy loss difference was +0.553 [−0.105, 1.316], both of which crossed zero. Given that 38 decision Sprints were used in multiple rounds of research and exploration, these intervals should be interpreted as descriptive comparisons under offline scenarios, and cross-project causal effects cannot be inferred from them. See Tables 9 and 10 for details.

Table 10. Paired descriptive comparisons with 38 Sprints as resampling units

Paired comparison (first − second)	Δ replanning incidence, percentage points [exploratory 95% interval]	Δ counterfactual proxy loss [exploratory 95% interval]
RF − fixed 20% reserve	−4.39 [−8.77, 0.00]	+1.105 [0.553, 1.684]
RF − calibrated Lag-1	−1.75 [−7.02, 3.51]	+0.553 [−0.105, 1.316]

Note: Source and statistical protocol: Comparisons use the main experiment file results/04_paired_sprint_bootstrap.csv. The 38 Sprints were resampling units, each retaining its three scenarios. We used 4,000 resamples with replacement, random seed 20261017, and exploratory 95% percentile intervals. Subsequent sensitivity profiles may have different interval versions; this table only uses the main experiment archive.

4.4.Project heterogeneity and discrete actions (RQ3)

Protocol E showed project-level variation. Compared with a fixed 20% reserve, the residual-adaptive rule produced lower proxy loss in 14 of 38 Sprints, identical loss in 16, and higher loss in 8; five Sprints could not meet the 50% simulated risk target at any available reserve. Its overall mean loss was 0.132 units higher, but project means differed: Aurora +1.000 (n=2), Mesos −1.500 (n=6), Spring XD +0.211 (n=19), and UserGrid +0.727 (n=11). Removing any single project could change the overall mean direction. These differences, together with unequal project sample sizes, limit the usefulness of a single pooled mean (Figure 3).

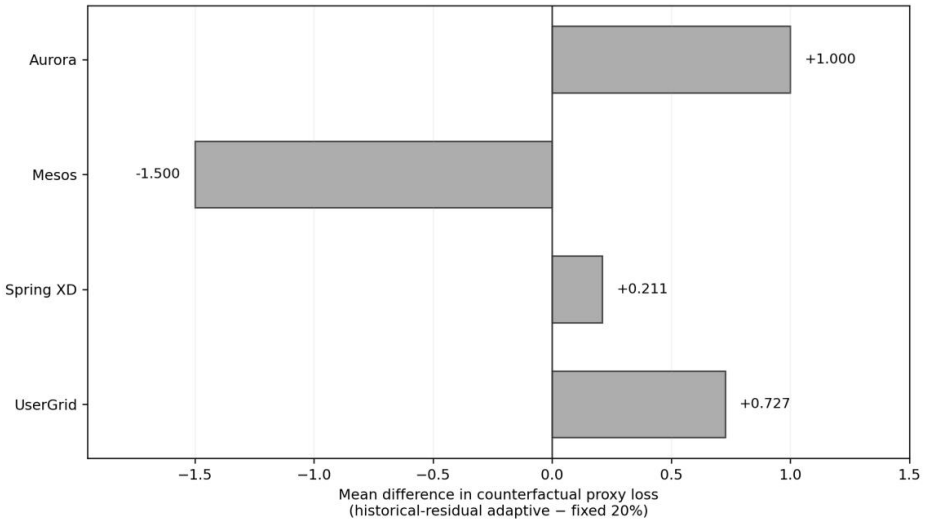

Fig 3. Protocol E: project-level mean counterfactual proxy-loss difference (historical-residual adaptive minus fixed 20%)

Note: Source: project-level simulation results from Protocol E; Aurora +1.000, Mesos −1.500, Spring XD +0.211, UserGrid +0.727. These results are distinct from the unified MILP experiment reported in Table 9.

4.5.Sensitivity analysis of capacity, workload, and value assumptions

The 54 sensitivity settings reuse the same 38 Sprints and vary capacity, arrival workload, priority-to-value mapping, and risk tolerance. At $\tau = 0.50$, RF met the simulated target in all 27 settings and had a lower incidence point estimate than the calibration-selected fixed policy in 24, but had lower counterfactual proxy loss in none. The $\tau = 0.75$ counts likewise show a risk–value trade-off. Figure 4 reports the

distribution of risk and proxy-loss effects across settings alongside the recorded counts such as 24/27 and 27/27.

A reserve policy should be assessed against its risk tolerance and the proxy-value cost of meeting that tolerance. At $\tau = 0.75$, the calibration-selected fixed policy met the target in 23/27 settings, compared with 27/27 for RF. A policy with lower proxy loss still fails the stated management requirement if its simulated risk exceeds the threshold. The original setting counts are retained in Appendix Table A1. Figure 4 presents effect-size distributions; the 54 settings do not constitute independent samples.

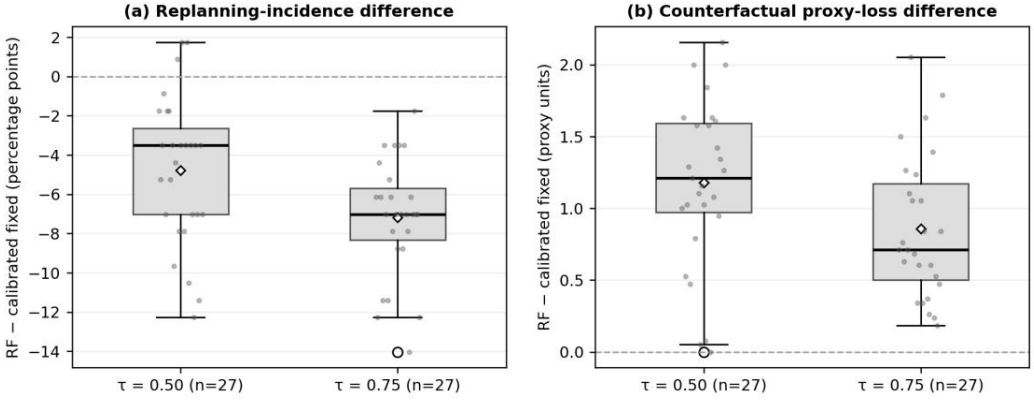

Fig 4. Distribution across 54 parameter settings of differences in simulated replanning incidence and counterfactual proxy loss between the random forest and calibration-selected fixed-reserve strategies

Note: Each risk-tolerance level contains 27 parameter settings, and the same 38 Sprints are reused. The comparator is the fixed reserve ratio selected during calibration, and not all settings are fixed at 20%. The boxplots summarize effect differences at the parameter-setting level and do not treat the 54 settings as independent samples; see Appendix Table A1 for the original setting count. Data source: sensitivity archive results/09_comparison_robustness_summary.csv.

4.6.Five illustrative sprint decisions

We selected five Sprints from the archived results to illustrate how reserve-ratio choices affect simulated triggers and counterfactual proxy loss. Each example uses the retrospective candidate pool and three synthetic shock scenarios. Table 11 reports the resulting modeled decisions, not observed interventions.

Table 11. Five sprint examples selected from the common-MILP results

Project / Sprint	Observed added tasks; RF / Lag-1 forecast	Reserved capacity: fixed → RF	Simulated triggers: fixed → RF	Counterfactual proxy loss: fixed → RF
Mesos 603	7; 13.7 / 1.0	20% → 30%	2/3 → 1/3	6 → 12
UserGrid 157	8; 11.9 / 2.0	20% → 30%	2/3 → 1/3	9 → 15
Spring XD 102	2; 12.1 / 15.0	20% → 15%	0/3 → 1/3	6 → 4
Spring XD 107	4; 4.1 / 0.0	20% → 5%	1/3 → 1/3	4 → 2
Aurora 174	N/A (risk guarantee cannot be inferred)	20% → 10%	3/3 → 3/3	1 → 1

Note: Source and nature: Five real sprint historical task attributes were selected purposefully based on the archive results; "Reservation ratio, number of simulated triggers, and value loss" are derived from offline synthetic scenarios rather than real project interventions.

For Mesos 603, RF selected a 30% rather than a 20% reserve: simulated triggers fell from 2 to 1 across three shocks, while counterfactual proxy loss rose from 6 to 12. Spring XD 102 shows the reverse trade-off, with less reserve and lower loss but one additional trigger. Spring XD 107 reduced its reserve without changing the trigger count. In Aurora 174, the minimum predicted risk was 57.66%, above the 50% target, showing that the fallback cannot guarantee compliance. Across 38 Sprints, RF had higher loss in 18 cases, equal loss in 18, and lower loss in 2; simulated triggers fell in 6, remained equal in 31, and rose in 1.

4.7.Paired risk – value classification and traceability audit

At $\tau = 0.50$, each of the 38 Sprints was classified by the paired changes in simulated replanning incidence (ΔR) and counterfactual proxy loss (ΔL), defined as RF minus fixed 20% reserve. One policy Pareto-dominates another if it is no worse on either measure and strictly better on at least one. The categories use archived values and apply to the shared retrospective task pool and simulated shocks.

For 18 Sprints, both measures were identical across the two policies. Fixed 20% Pareto-dominated RF in 12 Sprints; RF dominated fixed 20% in 1, and 7 showed a risk – loss trade-off. These classifications concern simulated comparisons, not observed replanning. They cannot establish the preferred policy for a team operating with its actual Sprint-start task pool (Table 12).

Each Sprint has only three simulated shocks, so the smallest nonzero change in its incidence estimate is one-third. The policies had identical trigger counts in 31 Sprints, partly reflecting this coarse scenario resolution. The paired categories describe behavior under the current model; assessing the same decisions in practice would require independent timestamped project records.

Of the 1,056 closing-report task-Sprint associations in 38 Sprints, only 247 can be traced to earlier reports; 15 Sprints have fewer than five traceable associations. These records do not reconstruct the task pool available at planning. Accordingly, the candidate pool is treated as a retrospective closing-report construct throughout the interpretation of policy comparisons.

Table 12. Exhaustive paired-outcome categories for 38 Sprints (RF minus fixed 20%; τ = 0.50)

Paired outcome category	Number of Sprints	Interpretation
Identical on both metrics	18	No difference under the three equally weighted stress scenarios
Fixed 20% Pareto-dominates RF	12	Fixed 20% is no worse on either metric and better on at least one
RF Pareto-dominates fixed 20%	1	RF is no worse on either metric and better on at least one
RF: lower risk, higher value loss	6	Risk reduction comes with higher opportunity cost
RF: higher risk, lower value loss	1	Opportunity-cost reduction comes with higher risk

Note: Complete paired classification of simulated triggers and counterfactual proxy loss for 38 Sprints under three synthetic shocks, not field-observed company performance.

4.7.1.Exploratory comparison by summary-level newly added task share

According to the AddedDuringSprint / IssueCountReported proportion of the summary reports at the end of each Sprint period, 38 decision Sprints were sorted and divided into 19 lower-share Sprints and 19 higher-share Sprints. Each group summarizes 57 existing simulated shock-scenario records, 114 in total; the prediction model and MILP strategy were not rerun. This grouping only reflects the added-task share in the summary report, and does not represent the proportion of new members in the candidate pool confirmed by timestamps. In the lower-share group, the random-forest-minus-fixed-20% difference in simulated replanning incidence was − 3.51 percentage points [− 10.53, 3.51], and the difference in the counterfactual proxy loss was +0.895 [0.316, 1.474]; the higher-share group was − 5.26 percentage points [− 10.53, 0.00] and +1.316 [0.421, 2.316]. These exploratory intervals describe the differences within each group, and the intervals do not establish a stable difference between the two groups (Table 13).

Table 13. Comparison of capacity reservation strategies grouped by report-level added-task share (τ = 0.50)

Summary-level added-share group	Reserve strategy	Sprints / scenarios	Simulated replanning incidence	Counterfactual proxy loss
Lower share	Fixed 20%	19 / 57	28.07%	5.526
Lower share	Historical distribution	19 / 57	42.11%	4.632
Lower share	Lag-1	19 / 57	28.07%	5.895
Lower share	RF	19 / 57	24.56%	6.421
Higher share	Fixed 20%	19 / 57	59.65%	6.421
Higher share	Historical distribution	19 / 57	59.65%	5.737
Higher share	Lag-1	19 / 57	54.39%	7.158
Higher share	RF	19 / 57	54.39%	7.737

Note: Sprints were ranked by added tasks / total tasks in closing summaries and split 19/19; each group contains 57 simulated records, not 57 independent Sprints. Group labels do not identify timestamp-confirmed additions.

4.8.Six-feature refit and external data feasibility

In the 72 later evaluation Sprints, RF6 had an MAE of 5.097, Lag-1 had 4.556, and the mean-of-last-three rule had 5.139. The RF6 and Lag-1 project MAEs were 3.583 and 4.583 for Aurora (n=12), and 3.049 and 2.724 for Mesos (n=29). They were 8.447 and 5.400 for Spring XD (n=20), and 6.056 and 7.818 for UserGrid (n=11). RF6 improved on RF7's 5.853 overall but remained worse than Lag-1, with inconsistent project rankings. RF6 is a post hoc forecasting diagnostic and is not combined with the main RF policy simulation.

The TAWOS audit assessed whether external records could support comparable event-time features. Among 11 projects outside the original four, 230 strict candidate Sprints were identified, but only five had explicit Sprint activation dates. Shifting the membership-change threshold from 0 to 24 hours after activation reduced recorded joins from 4,072 to 3,106, a 23.7% decrease. Of 3,668 Sprint summaries from 30 projects, 3,652 passed basic date/count checks and 3,488 had at least five completed historical Sprints; complete event histories nevertheless remained unverified. TAWOS therefore informs data-readiness and temporal-feature audits rather than a formal external prediction comparison[12].

A preliminary review of the Public Jira Dataset[13] found metadata concerning Sprint custom fields and issue-history changes, but the field dictionary and small archive prefix could not reconstruct a complete Sprint-start task pool or membership timeline. The dataset remains under structural feasibility assessment; a record-level audit is required before any external performance evaluation.

5.Discussion: forecast accuracy and planning trade-offs

5.1.Interpreting lower simulated risk

The candidate pool and attributes of the task-selection model come from the Sprint closing report. Therefore, tasks added after the planning meeting may appear in the simulated initial selection, even if they were not available at the actual planning time. Priority changes also cause a timing mismatch. According to $v = 6 - p$, if the numeric priority p of the task decreases after planning, that is, the priority increases, then the proxy score calculated from the closing value is higher than that calculated by using the earlier value. This judgment is only for the scoring direction of a single task, and cannot determine how the selected task set or the total counterfactual proxy loss changes. The existing data cannot count actual priority increases or determine how many tasks are added during the Sprint. The nominal capacity is calculated from the closing-report candidate pool story points, and the candidate pool and capacity scale are therefore linked. All strategies

share the same retrospective inputs to ensure the consistency of comparison conditions; if a timestamp-verified Sprint-start task pool is used, the selection results and MILP policy rankings may still change.

Added-task counts do not measure the associated workload. Two Sprints can each add five tasks yet require different story points or personnel effort. In the absence of reliable arrival-time estimates, our reserve-policy comparison uses specified synthetic workload scenarios. Complete event and estimation histories would allow forecast error and workload assumptions to be examined separately.

5.2.Practical implications for sprint planning

For Sprint planning, a reserve rule can be compared with fixed reservation and a simple historical baseline under the same candidate tasks, capacity, and simulated shocks. The first step is to specify an acceptable risk tolerance. The proxy value of work displaced by the reserve can then be compared with that risk estimate. Both outcomes should be reported. If every available reserve ratio exceeds the simulated risk target, the plan or resource allocation requires reconsideration. Applying this approach to actual planning also requires Sprint-start task states, membership timestamps, contemporaneous estimates, and observed team capacity.

5.3.Threats to validity

The scope of evidence in this study is mainly affected by three factors. First, the original export file lacks complete timestamped Sprint-start membership states and priority-change histories. We can calculate the proportion of new tasks added at the closing-report level, but we cannot confirm the proportion of tasks actually added in the MILP candidate pool, nor can we count the frequency of priority increases and their overlap with tasks added mid-Sprint. The formula $v = 6 - p$ can only explain the conditional direction of proxy-score change when a task's priority is raised. Second, the nominal capacity and candidate pool are calculated from closing-report story points, and the workload shocks and value mapping are also set by the research model; changing these assumptions may change the task-selection and relative proxy loss. Third, the 38 decision Sprints come from four projects, 19 of which belong to Spring XD. These samples have been analyzed many times in the process of method research, and 54 groups of sensitivity settings also use these 38 Sprints. Therefore, the results reflect the strategy differences under the specified offline scenarios, not observations of actual organizational outcomes.

5.4.Research contributions and scope of application

The framework links added-task forecasting with capacity reservation and a common mixed-integer task-selection model. It compares random forests with historical baselines on downstream simulated decisions rather than on forecasting error alone. Paired results for 38 Sprints describe how often policies agree or trade off risk against proxy value. Another 54 parameter settings show how the comparisons change with assumed capacity, arriving workload, and task value. The temporal audit also separates historical forecast features from retrospective task-selection inputs. Together, these analyses show why MAE alone is insufficient to evaluate reserve policies in the specified offline scenarios.

5.5.Scope of the paired diagnostics

Among the 38 paired Sprint comparisons, 18 had identical outcomes on both measures, fixed 20% dominated RF in 12, RF dominated the fixed rule in 1, and 7 involved trade-offs. This classification describes the retrospective candidate pool and synthetic shocks; it is not evidence of observed replanning or a policy ranking that holds across projects.

The available public records do not recover complete Sprint-start task states. We therefore interpret results from the repeatedly examined 38 decision instances as exploratory offline comparisons. Archived inputs, configurations, outputs, and checksums support inspection of the existing calculations. A new evaluation should fix the protocol in advance and use projects with reliable membership histories.

5.6.Reproducibility and decision-time evidence

The archived reproducibility audit records byte-identical outputs for six cleaned files, nine common-MILP CSVs, seven main sensitivity outputs, and two sensitivity-postprocessing files produced from the same four-project input archives. Two protocol JSON files differed only in elapsed time. The reviewer materials retain scripts, aggregate results, configurations, checksums, and a private archive of 72 paired predictions and MAE recalculation code. These reruns verify the archived computations, not actual team capacities or intervention outcomes.

Training, calibration, and evaluation were divided chronologically within each project. Forecasting uses summaries from completed Sprints, whereas the optimizer uses the target Sprint closing-report candidate pool. Historical feature timing is therefore documented more directly than the timing of task membership in the optimizer. A planning-time evaluation would require Sprint-start task lists, addition times, contemporaneous priorities and story points, and measured team capacity.

The same 38 later Sprints were used during method development, and the 54 sensitivity settings draw on those same instances. Their risk and proxy-loss values describe the specified simulation protocol. They do not measure observed Sprint commitments or economic benefits. Independent evaluation requires complete event histories and a comparison protocol defined before examining the test data.

5.7.Data requirements for cross-project validation

The S7A – S7G external audit examined whether historical predictors could be reconstructed in other datasets and whether Sprint event timelines were sufficiently reliable. Excluding last_dev resolved one feature-alignment issue, but the definition of added tasks and the completeness of event records still require verification. Similar feature distributions are preliminary evidence of comparability, not transfer performance. External evaluation should fix models and label

definitions before selecting projects and use verified histories to compute MAE and decision outcomes.

Reserve policies should be assessed by reporting simulated replanning risk alongside the proxy value of work deferred. Neither lower forecasting error nor a lower risk point estimate automatically establishes a superior plan. Testing organizational applicability requires planning-time task membership, priorities, workload, and actual team capacity.

6.Conclusions and management implications

This paper uses 238 Sprints from four public Jira projects to analyze how predictions of the number of added tasks affect capacity reservation and task-selection. In 72 later-evaluation samples, the MAE of the seven-feature random forest RF7 was 5.853, Lag-1 was 4.556, and the six-feature retrained model RF6 was 5.097. The unified offline comparison for 38 Sprints showed that at a simulated risk tolerance of 50%, the simulated replanning-incidence point estimate for the random forest strategy was 39.47%, compared with 43.86% for fixed 20% reservation; the exploratory 95% interval for the paired difference of −4.39 percentage points was [− 8.77, 0.00], so a consistent risk reduction is not established. At the same time, the counterfactual proxy loss calculated based on closing-report task attributes increased from 5.974 to 7.079, and the paired difference was +1.105, with an interval of [0.553, 1.684]. Relative to calibrated Lag-1, the paired intervals for risk and proxy loss both crossed zero.

The comparison between the two groups based on the proportion of new tasks in the final report and 54 sensitivity settings further presents the trade-off between simulated risk and the proxy value that can be scheduled. Since the candidate pool and nominal capacity are from the Sprint final report, the conclusions apply to the specified counterfactual scenarios rather than directly measured organizational performance. When actually selecting the reserved proportion, in addition to referring to the prediction of new tasks, the choice must also consider the team's risk tolerance and the proxy value of delayed tasks. Subsequent studies need to include the initial task status, task attribute change history and actual team capacity with time stamps, and use independent project data for evaluation.

Author contributions Yushuai Cong is the sole author of this article and is responsible for all aspects of the work: study conception and design, data collection and analysis, and drafting and revision of the manuscript.

Funding No funding was received for this work.

Data availability The data that support the findings of this study are available from the corresponding author upon reasonable request. The classification of supplementary materials and reproducibility verification details are provided in Appendix B.

Declarations

Conflict of interest The author declares no conflict of interest.

AI Use Statement During manuscript preparation and revision, the author used OpenAI ChatGPT (GPT-5.6 Sol) to assist with academic language editing, manuscript organization, code development, statistical verification and responses to peer-review comments. The author checked and revised the relevant content and code with reference to the original datasets, experimental records and cited literature. AI-generated content was not used as independent research evidence; the numerical results and conclusions reported in the paper were checked against the study's evidence. The author assumes full responsibility for the accuracy, originality and academic integrity of the paper.

References

[1] ELMACHTOUB A N, GRIGAS P. Smart “Predict, then Optimize”. Management Science, 2022, 68(1): 9 – 26.
[2] ÖZÇELİKKAN N, TUZKAYA G, ALABAŞ USLU Ç, et al. A multi-objective agile project planning model and a comparative meta-heuristic approach. Information and Software Technology, 2022, 151: 107023.
[3] GOLFARELLI M, RIZZI S, TURRICCHIA E. Multi-sprint planning and smooth replanning: An optimization model. Journal of Systems and Software, 2013, 86(9): 2357 – 2370.
[4] BREIMAN L. Random forests. Machine Learning, 2001, 45: 5 – 32.
[5] MANDI J, KOTARY J, BERDEN S, et al. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. Journal of Artificial Intelligence Research, 2024, 80: 1623 – 1701.
[6] GIBBS I, CANDÈS E J. Adaptive conformal inference under distribution shift. Advances in Neural Information Processing Systems, 2021, 34: 1660 – 1672.
[7] CERQUEIRA V, TORGO L, MOZETIČ I. Evaluating time series forecasting models: An empirical study on performance estimation methods. Machine Learning, 2020, 109: 1997 – 2028.
[8] CAWLEY G C, TALBOT N L C. On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research, 2010, 11: 2079 – 2107.
[9] KORALAGE R. AgileScrumSprintVelocityDataSet [Data repository]. GitHub. URL: https://github.com/RandulaKoralage/AgileScrumSprintVelocityDataSet (accessed October 17, 2026).
[10] ABEBE M, RUSIECKI A. Itemlet Dataset: A Multi-project, Multi-domain, and Feature-engineered Dataset for Empirical Software Engineering [Data set]. Zenodo, 2026. DOI: 10.5281/zenodo.19411554 (accessed October 19, 2026).
[11] ABEBE M, RUSIECKI A. A multi-project, multi-domain, and feature-engineered Itemlet dataset for empirical software engineering. Scientific Data, 2026. DOI: 10.1038/s41597-026-08164-5.
[12] TAWOSI V, AL-SUBAIHIN A, MOUSSA R, et al. A versatile dataset of agile open source software projects//Proceedings of the 19th International Conference on Mining Software Repositories (MSR). 2022: 707 – 711.
[13] MONTGOMERY L, LÜDERS C, MAALEJ W. The Public Jira Dataset (v7) [Data set]. Zenodo, 2025. DOI: 10.5281/zenodo.15719919 (accessed October 19, 2026).

Appendix A. Counts from the sensitivity settings

The original counts are retained alongside the Figure 4 effect-size distributions. All 54 parameter settings reuse the same 38 Sprints and are not independent tests.

Appendix Table A1. Settings in which RF improved on the calibration-selected fixed-reserve policy

Risk τ	Settings	RF lower risk	RF lower loss
τ = 0.50	27	24	0
τ = 0.75	27	27	0

Note: Counts describe parameter sensitivity using the same 38 Sprints and do not constitute independent significance tests.

Appendix B. Supplementary materials and reproducibility verification

B.1 Supplementary materials list

The supplementary materials are organized as S1 (scripts and execution steps), S2 (aggregate prediction and policy results), S3 (protocols and time-validity audits), and S4 (verification records and source-license notes). The full contents of S1 – S4 are not included in this publication, and are available from the corresponding author upon reasonable request. MAE interval recalculations, project-level temporal and coverage audits, the 72 matched prediction records and the MAE bootstrap script are also available from the corresponding author upon reasonable request. The release of source-linked derivatives remains subject to assessment of applicable redistribution terms.

B.2 Reproduction verification records and data scope

Raw data and samples: The SHA-256 of the original ZIPs for the four projects is adecbbb3b2264d53414289b19f9e2cf6fb976424d7a2ea5de4a0b84cfd604751; the SHA-256 of the third-stage cleaned archive is 9f77ef4c9bd440735d65d00ce2b5073dd655889e0d0584b27a9c960edabcfe76. The main comparison and sensitivity analysis used the exact same cleaned archive: 238 valid Sprints, 4,726 deduplicated task-Sprint associations; 129 training, 37 calibration, and 72 late-prediction samples, of which 38 were used for planning simulations under three synthetic stresses.

Consistency of reruns: The archived rerun audit compared 24 numerical outputs: 6 cleaned files, 9 common-MILP outputs, 7 main sensitivity outputs, and 2 sensitivity-postprocessing outputs. They matched the saved outputs byte for byte. The only differences in the protocol JSON files involved elapsed time. This verifies repeatability of those calculations on the archived inputs; it does not measure actual team capacity or the effects of an intervention.

Statistical protocol and model version: Table 10 uses 38 Sprint-level resampling units with their three shocks clustered in 4,000 draws (seed 20261017). The prediction MAE intervals in Table 6 instead use 72 matched Sprint records, project-stratified paired resampling with fixed project counts, 8,000 draws, and seed 20261111. Sensitivity postprocessing used a separate 12,000-draw procedure and does not replace the main Table 10 intervals. The post hoc RF6 refit used the original 129 training Sprints; its later MAE was 5.097, and archived versus rerun RF6 predictions differed by at most 3.55×10^{-15}.

The forecast target is the number of added tasks recorded at Sprint close, not their true story-point workload. Decisions use historical task attributes plus explicit synthetic capacity, shock, and proxy-value assumptions. TAWOS and Public Jira Dataset audits assessed data readiness but did not yield certified complete membership histories or independent prediction tests.