Effort-Based Agile Release Forecasting Without Story Points:
Institutionalizing a Percentile-Based Decision Routine with Uncertainty Propagation

Berk Kibarer — berkkibarer@gmail.com
Independent Researcher

Manuscript for the *In-Practice* track | January 2026

Abstract

In many product organizations, release commitments ultimately need to be expressed in calendar time. Yet in sprint-based delivery, teams often spend substantial effort debating relative size (e.g., story points) and then struggle to translate velocity into stakeholder-facing dates. In this manuscript, we present a story-point-free forecasting routine that uses direct effort logging (person-days) and a *sequential simulation (Monte Carlo) loop* to produce percentile-based forecasts (P10–P90). The key contribution is not a statistical novelty; rather, it lies in the institutionalization of a percentile-based commitment policy within a multi-year operational routine. We report how this forecasting approach was operationalized and sustained as a weekly governance mechanism over approximately 170 weeks in a large-scale, product-based organization in the communications sector.

We describe (i) the practical context that motivated the work, (ii) the export-based pipeline that converts Jira/Trello data into a recurring report, (iii) a commitment policy anchored to P90, and (iv) adoption frictions and mitigation strategies. Because publishable per-forecast industrial logs were not available after the author left the organization, we present operational evidence using coarse, confidentiality-preserving proxies. Across 16 releases, measured as schedule slip relative to the *final P90 commitment* prior to release, delivery clustered around days rather than weeks: 3/16 finished approximately 3–4 days early, 8/16 were delivered with 0–3 days delay, 3/16 with 3–5 days delay, and 2/16 with 5–10 days delay. In a pre-adoption baseline of 3 releases, schedule slip ranged approximately between 5–8 weeks per release. We complement the field report with synthetic behavioral checks to validate expected model behavior under varying history length and volatility without claiming numerical equivalence to the industrial slip buckets.

Keywords: industrial experience report; agile forecasting; #NoEstimates; Monte Carlo simulation; uncertainty; governance; release planning; effort tracking

1. Introduction

This section motivates the manuscript from the perspective of a practitioner-reader: what problem appeared in day-to-day release planning, what changed after adoption, and what this paper contributes to the In-Practice community.

In the studied context, a product team repeatedly faced a familiar but costly pattern: stakeholders asked for a date, while the team's planning meetings gravitated toward calibrating relative size. The practical issue was not that story points are inherently wrong; rather, they introduced friction in communication and decision-making. Conversations about risk were frequently displaced by conversations about calibration.

The team needed a routine that (a) speaks in calendar time, (b) makes uncertainty explicit, and (c) is lightweight enough to run weekly without becoming another process burden. The outcome was a story-point-free forecasting routine: effort is logged as person-days; a pipeline extracts metrics from the work system; and a simulation generates a forecast distribution used to drive decisions. The most important institutional change was a governance rule: external commitments are anchored to a conservative percentile (P90) rather than a single "most likely" date. The routine described in this manuscript was applied over approximately 170 weeks of operational use, spanning 16 release cycles in the observed policy era.

1.1 What this manuscript claims (and does not claim)

This manuscript does *not* claim a new Monte Carlo method. Monte Carlo forecasting is well-known in agile planning. The novelty lies in documenting the **institutionalization of a percentile-based decision regime** under enterprise constraints: the data pipeline, the weekly operational routine, the governance policy, and the sociotechnical adoption over multi-year use. We also do not claim statistically controlled accuracy

improvements, as detailed industrial forecast logs are not publishable. Instead, we provide confidentiality-preserving operational evidence and a transparent discussion of limitations.

1.2 Contributions for the In-Practice readership

- **Operational routine:** a repeatable weekly process from data export to stakeholder-facing forecast artifacts.
- **Governance policy:** commitment anchored to P90 (tail-risk management) rather than negotiated "single dates."
- **Implementation blueprint:** minimal data requirements, pipeline structure, and how-to guidance.
- **Field texture:** adoption friction, what failed early, what was adjusted, and how compliance was enforced.
- **Evidence architecture:** how to report value without disclosing sensitive logs, and how to avoid over-claiming.

2. Background

This section introduces the minimal concepts required to understand the rest of the manuscript: story points vs. effort, percentile forecasts, and why uncertainty matters in release commitments.

2.1 Story points and the communication gap

Story points are commonly used for relative estimation and velocity tracking. In practice, teams often face two recurring issues: (i) calibration drift over time and across team changes, and (ii) translation friction from abstract points to stakeholder-facing calendar commitments. Practitioner discussions highlight these drawbacks and motivate alternative measurement approaches (e.g., #NoEstimates).

2.2 Percentile forecasts as risk signals

A probabilistic forecast provides a distribution of completion outcomes rather than a single number. Percentiles (P50, P90) are particularly practical: P50 represents the median (a central tendency), while P90 provides a conservative date intended for commitments under uncertainty. The practical utility of percentiles depends less on the exact statistical model than on whether the organization uses the signal consistently to drive decisions and trade-offs.

3. Related work and gaps

This section positions the work against prior practice and literature, and makes explicit the gaps this manuscript addresses. We keep the discussion intentionally tight: In-Practice readers do not benefit from an extended literature war.

3.1 Practitioner foundations

Monte Carlo forecasting for agile planning is widely discussed in practitioner-oriented sources, often using historical velocity (e.g., story points) as an input. Similarly, #NoEstimates emphasizes measurement over estimation overhead and argues for empirical approaches to planning. These foundations are valuable but often under-report how forecasting becomes a sustained organizational routine: who runs it, how it integrates with work systems, and how commitment policies are enforced over time.

3.2 Academic anchors on uncertainty and estimation

The broader software engineering literature emphasizes uncertainty as a dominant factor in software planning and argues that estimation and planning methods must be evaluated under realistic constraints rather than idealized assumptions. Evidence syntheses in software cost estimation further highlight wide variability in estimation performance and the importance of transparent assumptions and reporting (Jørgensen & Shepperd, 2007).

3.3 Gaps this manuscript closes

- **Governance gap:** prior Monte Carlo discussions rarely document *commitment policies* (e.g., P90 anchoring) as institutional rules.
- **Operationalization gap:** #NoEstimates emphasizes measurement but often lacks export-based, repeatable pipelines and artifacts for release planning.

- **Longitudinal gap:** many discussions present methods but provide limited detail on multi-year sustainment, adoption friction, and enforcement mechanisms.
- **Reporting gap:** guidance is limited on how to present value credibly when detailed industrial logs cannot be published.

4. Practical context and setting

In-Practice readers typically want to know early: where did this problem arise, what constraints shaped the solution, and what kind of team and release cadence should they imagine. This section provides that context with strong anonymization.

4.1 Organization and domain

The routine was developed and used within a large-scale, product-based organization in the communications sector. The team owned a central test-automation platform that served multiple internal customers across digital services. The organization is anonymized for commercial confidentiality; we report only aggregated descriptors relevant to adoption and operation.

4.2 Team structure and cadence

The team typically ranged between 9–12 members, with a sustained core around 11 (developers, testers, and developer-in-test roles). Work was managed in weekly feedback cycles. For reader simplicity we use the term "sprint" to refer to these weekly cycles, as the data collection and reporting rhythm matched a one-week sprint cadence. Releases occurred roughly every 2–3 months, spanning approximately 16 releases in the observed policy era.

4.3 Tooling and data sources

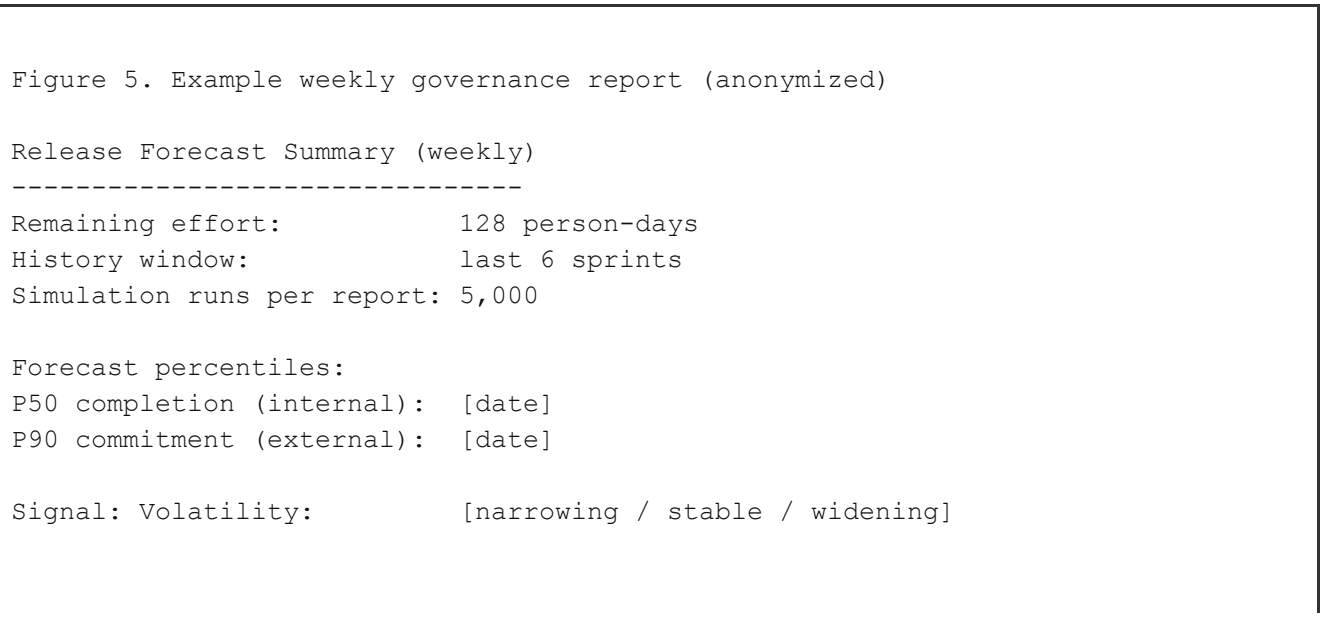
Work management used Jira for backlog tracking (multiple boards) and Trello for operational bookkeeping and effort logging. A lightweight integration extracted and normalized fields from Jira/Trello into a consistent dataset for forecasting. Data extraction and preprocessing were automated; a weekly governance step focused on review, interpretation, and decisions.

5. What we did: weekly routine, pipeline, and governance

This section is the operational core of the manuscript. A reader should be able to understand the routine and replicate it even without fully internalizing the underlying statistics.

5.1 Weekly routine (who does what, and how long it takes)

The routine ran weekly. Data ingestion from Jira/Trello and metric computation were automated. A small governance step—performed by the author and one analyst—reviewed data quality, interpreted the report, and extracted actionable lessons. The overall weekly cycle (including governance reflection) typically completed within **about 1 hour**. This hour included not only "data collection" but also process feedback evaluation and routine adjustments.



```
Action: If P90 moves out:   trim scope / allocate buffer / stabilize
```

Figure 5: Simplified excerpt of the weekly governance artifact. The paper's contribution is the institutionalization of this recurring report-and-decision routine rather than a claim of statistical novelty. A full, runnable version of the pipeline and a sample report artifact are provided in the replication package (see Data & Code Availability).

5.2 Pipeline overview (export-based operationalization)

The pipeline was intentionally designed to minimize friction: it relied on exports and a stable schema rather than heavy customization in the work tracking tools. Figure 1 summarizes the pipeline at a glance.

```
Figure 1. Forecasting pipeline (schematic, anonymized)

[Jira Issues]        [Trello Effort Logs]
     |                      |
     +---- export/API -------+
                |
          [Normalization]
   (required fields checks,
    mapping, deduplication)
                |
         [Metric Computation]
   (throughput, unplanned work,
    volatility, health signals)
                |
     [Forecast Simulation]
   (percentiles P10..P90,
    calendar-aware capacity)
                |
        [Single-page Summary]
     (executive view + guidance)
                |
         [Weekly Governance]
  (interpretation, decisions,
   scope/buffer/capacity actions)
```

Figure 1: The export-based pipeline from work systems to a weekly forecast artifact and decision routine.

5.3 Data discipline rules

Adoption required a small amount of enforceable data discipline. In practice, this meant (i) a weekly cut-off (e.g., Monday), (ii) dashboard visibility for missing or inconsistent entries, and (iii) a short governance review where gaps were corrected so the report could be used for decisions. The concrete checklist of required fields and enforcement rules is summarized in Table 1.

5.4 Minimal required fields (implementation checklist)

To reduce confidentiality risk while still enabling replication, Table 1 lists the minimal fields that were required in the weekly export to run the routine. Organizations can extend these fields, but the routine was intentionally designed to work with a small, stable schema.

Data element	Where it came from	Why it is required	Operational rule / enforcement
Work item identifier	Jira issue key	Deduplication; stable joins across exports	Required in export; missing keys rejected in normalization
Status / done indicator	Jira workflow state	Compute completed effort per sprint	Weekly cut-off; dashboard highlights items stuck in inconsistent states
Completion date (or sprint assignment)	Jira sprint field / resolution date	Align completion to weekly cadence	Cut-off applied weekly; exceptions reviewed in governance step
Effort logged (person-days)	Trello log (or equivalent effort ledger)	Primary signal replacing story points	Logged weekly; reviewed and forced for completeness when missing

Unplanned work marker	Jira label / type / custom field	Compute disruption (unplanned fraction)	Governance review of misclassified items; feedback to team
Optional: bug/quality proxy	Jira issue type / tags	Approximate “quality tax” component	Used as a coarse percentage; not used for individual evaluation

Table 1: Minimal required fields for the export-based routine (anonymized and implementation-focused).

5.5 Commitment policy (the decision mechanism)

The most important institutional change was a policy shift: **external commitments were anchored to P90**. The forecast distribution was used as a risk signal. When the P90 date moved unfavorably or uncertainty widened, decisions were made through scope trade-offs, buffer windows, or capacity interventions rather than by negotiating a single date. Internally, teams may use the median (P50) as a planning reference; however, the key institutionalized rule in this context was the stakeholder-facing commitment anchor at P90.

5.6 Common failure modes during institutionalization (and mitigations)

Early in adoption, the team encountered repeatable failure modes that required explicit mitigation. These details matter for transferability because they describe what tends to break before a forecasting routine becomes institutional.

- **Ignored artifact (low consumption):** early reports were produced but not consistently used in planning. *Mitigation:* introduce a one-page executive summary and explicitly connect the report to the weekly planning agenda.
- **Percentile resistance (“P90 is too pessimistic”):** some stakeholders preferred tighter dates. *Mitigation:* standardize the commitment anchor at P90 and treat deviations as tail-risk signals that trigger scope/buffer discussions.
- **Data discipline drift (missing or late effort logs):** effort entries occasionally lagged behind delivery reality. *Mitigation:* weekly cut-off (e.g., Monday), dashboard visibility, and forced correction as part of the governance review.
- **Category gaming risk:** temptation to reclassify work to “look better.” *Mitigation:* keep metrics at team level; explicitly prohibit individual ranking; review outliers qualitatively.

5.7 Longitudinal operation: how the routine stabilized over time

Operational indicators of institutionalization included consistent weekly report generation, explicit use of the P90 forecast in stakeholder-facing release commitments, and the integration of forecast review as a standing agenda item in release planning meetings. In this setting, the routine was broadly internalized after roughly the first three release cycles—about 9 months given a 2–3 month cadence—after which the weekly process and percentile language became standard in release conversations.

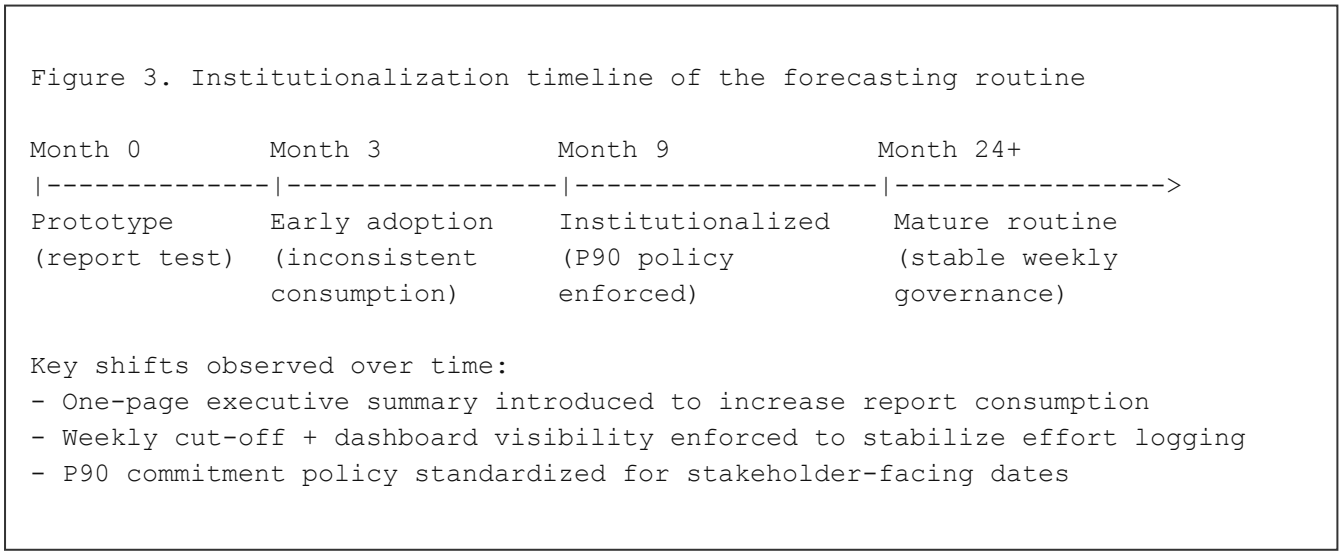


Figure 3: Timeline of how the routine moved from prototype reports to an institutionalized, policy-driven weekly governance practice.

6. Method: what the simulation does and why

This section explains the forecasting method at a practitioner-appropriate level: inputs, outputs, and why we model uncertainty. The heart of the manuscript is the routine and governance, not the model; therefore, we keep mathematics out of the main narrative and place the equation in Appendix A.

6.1 Inputs and outputs

Inputs: historical weekly effort completion, unplanned work fraction, and optional quality indicators (e.g., bug share), plus calendar constraints (weekends/holidays).
Outputs: a forecast distribution of completion dates, summarized as percentiles (P10, P25, P50, P75, P90) plus diagnostic signals (e.g., volatility/disruption).

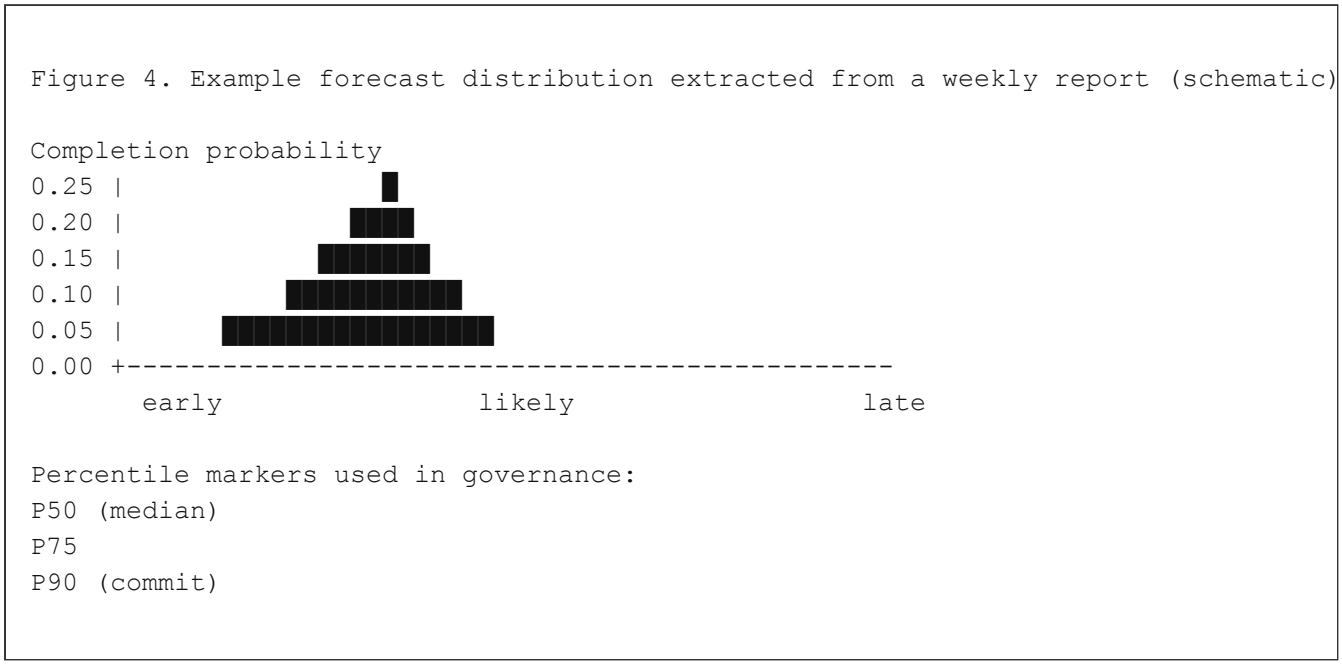


Figure 4: Schematic forecast distribution produced by the simulation loop. The weekly routine operationalizes percentiles as decision signals; in the studied context, the stakeholder-facing commitment anchor is P90.

6.2 Two sources of uncertainty (practical rationale)

We separate uncertainty into two layers because they behave differently in practice: (i) **parameter uncertainty** (limited history makes fitted parameters unstable) and (ii) **residual variability** (delivery noise: incidents, absences, scope churn). This separation makes forecast width easier to interpret: wide intervals may come from low history (parameter instability) or from high volatility (process noise), and the governance response differs accordingly.

6.3 Sequential simulation loop (high-level description)

The simulation proceeds sprint-by-sprint (weekly), sampling a plausible daily throughput rate for each future sprint, subtracting capacity from remaining effort, and stopping when work completes. Calendar constraints are applied so that forecast dates remain realistic.

In each simulation run, we first sample model parameters to propagate *parameter uncertainty* (drawing β from the estimated covariance when available) and then sample a residual term ε to represent *process noise*. For each simulated sprint, we also sample a historical feature row (with replacement) from the recent window to reflect observed operating conditions (e.g., unplanned work fraction and other feature values), and compute a non-negative daily throughput as $\max(\beta^T x + \varepsilon, 0)$.

The implementation uses thousands of simulation runs to produce stable percentile estimates. In the industrial routine, we used **5,000 runs** per weekly report and a **rolling history window of the most recent 6 sprints**. Sensitivity checks indicated that percentile estimates stabilized beyond 1,000 runs.

The window length was chosen empirically to balance stability and responsiveness: shorter windows (<4 sprints) produced unstable percentile estimates, whereas longer windows (>10–12 sprints) diluted recent delivery conditions and increased forecast drift.

The goal is not methodological sophistication; it is a repeatable, interpretable signal that can be institutionalized in governance. A minimal throughput functional form used in the implementation is provided in Appendix A for transparency.

7. Evidence: operational outcomes and behavioral checks

In an In-Practice manuscript, evidence is often a combination of operational outcomes, repeatability of the routine, and transparent limitations. This section reports what can be supported without disclosing sensitive industrial logs.

7.1 Schedule slip metric (definition and timestamp)

Schedule slip is defined as the difference between the actual release date and the **final forecasted commitment date (P90)** that was communicated prior to release. Negative values indicate early completion.

Timestamp clarification: we define the *final P90 commitment* as the P90 date in the **last weekly report produced before the actual release date** (typically about 1 week prior). If the planned release date moved during delivery, the "final commitment" reflects the organization's last communicated P90 commitment before release, i.e., the last published weekly report before the actual date.

Note on units: the organization internally tracked slip primarily in business days; in this manuscript we report coarse day-scale buckets (and week-scale baseline buckets) because the difference is not decision-relevant at the presented granularity.

7.2 Operational proxies (field evidence without publishable logs)

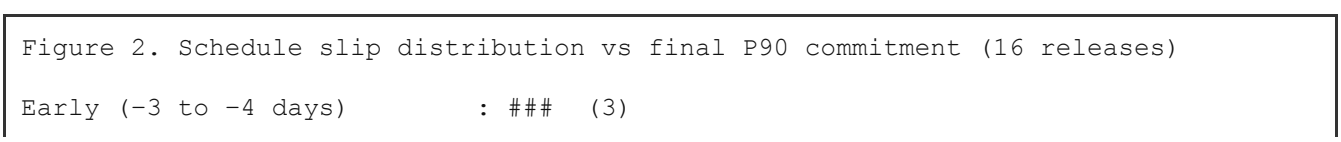
Operational proxy	What we can report (coarse, confidentiality-preserving)	Why it matters
Observed schedule slip distribution (policy era)	Across **16 releases**, measured as schedule slip relative to the final P90 commitment: • 3/16 finished about 3–4 days early (negative slip) • 8/16 delivered with 0–3 days delay • 3/16 delivered with 3–5 days delay • 2/16 delivered with 5–10 days delay Reported as coarse buckets to avoid false precision in the absence of publishable per-forecast logs.	Replaces "soft" claims with an explicit denominator, a precise metric definition, and interpretable buckets.
Pre-adoption baseline (comparison)	In the pre-adoption period (**3 releases**), schedule slip ranged approximately between **5–8 weeks** per release. Baseline releases occurred within the same product mission context and comparable release horizons (about 2–3 months), although staffing levels and operational pressures varied over time.	Provides a coarse "before/after" comparison while explicitly flagging confounds and avoiding causal claims.
Decision mechanism shift (policy fact)	External commitments were anchored to **P90 by policy** using the final weekly report prior to release; this was consistently applied across the release cycles in scope.	Converts "decision change" claims into governance facts, reducing speculative counterfactual reasoning.
Operational effort for running the routine	Automated data extraction and computation; weekly governance review and decision extraction typically completed in about 1 hour by two roles (author + analyst).	Answers: "What does it cost to run this weekly?" and supports transferability.

Table 2: Operational proxies used to report value under confidentiality constraints.

Because external commitments were intentionally anchored to P90, the operational objective of the forecasting routine was not exact-day prediction but the management of tail risk in release commitments. Operational success was therefore evaluated by whether delivery remained within the expected tail envelope (days rather than weeks), rather than by eliminating schedule slip entirely.

7.3 One compact visual: release slip distribution (policy era)

Reviewers often respond well to a single visual summary. Figure 2 is a confidentiality-preserving histogram-like view using the same buckets as Table 2. It does not reveal per-release dates; it only visualizes counts.



```
0–3 days delay                : ########  (8)
3–5 days delay                : ###  (3)
5–10 days delay               : ##  (2)
```

Figure 2: Bucketed schedule slip counts (policy era). Negative values indicate early delivery.

7.4 Vignettes (Situation–Signal–Decision–Outcome, with one numeric proxy each)

The vignettes illustrate repeatable decision patterns rather than claiming controlled causal effects. To reduce the "anecdotal" attack surface, each vignette includes a single numeric proxy (reported as a coarse range).

Vignette 1 — Buffering a commitment using P90

- **Situation:** Stakeholders requested a fixed date for a release window spanning multiple services.
- **Signal:** The weekly report showed the P90 commitment drifting later by approximately **1–2 weeks** over consecutive reports (e.g., from 6 to 8 weeks remaining), while the interval width widened.
- **Decision:** The team communicated the P90 date as the commitment, scheduled an explicit buffer window, and trimmed lower-value scope.
- **Outcome:** Delivery remained within the day-scale slip buckets relative to the final P90 commitment (i.e., within **≤10 days**), with reduced escalation pressure due to the explicit buffer.

Vignette 2 — Volatility-triggered stabilization

- **Situation:** A period of elevated unplanned work and operational interruptions increased forecast width.
- **Signal:** The unplanned work fraction increased noticeably (operationally interpreted as "high volatility"), and the P90 commitment moved outward by roughly **1 week** between weekly reports (e.g., from 7 to 8 weeks remaining).
- **Decision:** The team deferred optional features, prioritized stabilization and infrastructure hardening, and restored discipline in logging and triage.
- **Outcome:** Over subsequent weeks the forecast interval narrowed (governance signal improved), enabling renewed commitments under the same P90 policy.

Vignette 3 — Capacity shift (turnover) and conservative governance

- **Situation:** Turnover and role changes temporarily reduced effective capacity while external dates remained under pressure.
- **Signal:** The report showed a sustained reduction in effective throughput (interpreted operationally as lower capacity), with P90 moving outward by approximately **1–2 weeks** across consecutive weekly reports (e.g., from 6 to 7–8 weeks remaining).
- **Decision:** The team maintained the P90 commitment policy, re-scoped work into "must-have" and "deferrable" items, and made the buffer explicit in stakeholder communication.
- **Outcome:** The routine preserved situational awareness and prevented silent optimism; delivery stayed in day-scale slip buckets under the conservative commitment policy.

7.5 Synthetic behavioral checks (not numerical equivalence)

To avoid over-claiming from limited industrial evidence, we complement the field report with controlled synthetic experiments. Synthetic experiments are used as behavioral checks under controlled conditions: (i) low history yields wider intervals due to parameter uncertainty, (ii) higher volatility yields wider intervals due to process noise, and (iii) beyond 20–24 sprints, additional history shows diminishing returns while increasing drift risk. We do not claim numerical equivalence between synthetic forecast distributions and industrial schedule-slip buckets, as they summarize different observables. The purpose of the synthetic checks is not to validate industrial outcomes numerically, but to verify that the forecasting pipeline behaves consistently with expected uncertainty dynamics, such as sensitivity to history length and volatility.

8. Discussion: what changed, when it fits, and when it fails

This section answers practitioner questions reviewers and readers will ask: did decisions truly change, is the change repeatable, and in which contexts should (or should not) this routine be adopted.

8.1 What routine changed (and why that matters)

Three shifts mattered operationally: (1) a weekly export-based pipeline replaced ad hoc status narratives, (2) a one-page executive summary increased report consumption, and (3) a governance policy anchored commitments

to P90. The combination moved conversations from “calibrate size” toward “manage tail risk,” using scope, buffers, and capacity as the primary levers.

8.2 Did decisions change in a repeatable way?

We do not present a controlled counterfactual; instead we report repeatable decision patterns anchored in governance facts. After adoption, the commitment anchor (P90) was fixed by rule, and when forecasts worsened the typical responses were scope trimming, buffer window allocation, or stabilization work. This was materially different from earlier practice where commitments tended to be communicated without an explicit tail-risk policy.

8.3 Where this approach fits

- Teams under sustained date pressure where stakeholders need calendar-time commitments.
- Contexts where effort logging is feasible and can be enforced with minimal rules.
- Environments where uncertainty discussions are preferred over calibration debates.
- Organizations facing tool friction where heavy customization is unrealistic.

8.4 Where this approach fails or should not be used

- Teams unwilling or unable to maintain effort logging discipline.
- Highly exploratory R&D work where effort observability is extremely low.
- Settings where metrics are used for cross-team ranking or individual performance control (misuse risk).
- Very low-history situations where any percentile policy would be dominated by uncertainty (use conservative buffers until history accumulates).

9. Threats to validity and limitations

This section is explicit about what the evidence can and cannot support. Clear limits increase reviewer trust.

9.1 Confidentiality and missing publishable industrial logs

The industrial partner is anonymized, and detailed per-forecast logs are not publishable after the author’s departure. Even with full logs, the primary contribution of this manuscript would remain the operational integration and governance policy, rather than a claim of statistical superiority. We therefore report value through denominator-based operational buckets and a transparent description of routine, roles, and enforcement mechanisms.

9.2 Observational comparisons (not statistically controlled)

The pre-adoption vs. policy-era comparison is descriptive and not statistically controlled. We avoid causal claims and focus on operational plausibility, repeatability of the routine, and policy facts. Confounds (e.g., staffing and external pressure) are acknowledged explicitly.

This manuscript reports the experience of a single team in a specific organizational context. The intended contribution is therefore analytical generalization—a transferable operational routine and governance pattern—rather than statistical generalization across organizations.

9.3 Synthetic data limits

Synthetic behavioral checks cannot reproduce all real-world dependencies and shocks. They are used to validate expected pipeline behavior under controlled volatility and history-length conditions, not to claim industrial numerical accuracy.

9.4 Metric misuse risk

Effort metrics can create perverse incentives if used for cross-team comparison or individual ranking. Adoption should include explicit policy and education to keep usage at the level of team trend analysis and uncertainty management.

10. Conclusion

This manuscript documents not a forecasting technique, but the **institutionalization of a percentile-based decision regime** for release planning under enterprise constraints. The practical method replaces story points

with person-day effort logging, produces a weekly forecast distribution, and anchors external commitments to P90 by policy. Over sustained operation, the routine provided a repeatable way to discuss uncertainty and manage tail risk through scope, buffers, and capacity interventions.

Operational evidence is reported through confidentiality-preserving proxies. Across 16 releases measured against the final P90 commitment, schedule slip clustered in day-scale buckets (including 3 early releases and most deliveries within a few days of the commitment), while a pre-adoption baseline of 3 releases exhibited week-scale slips (5–8 weeks). We present these outcomes as observational and non-controlled; the core value for the In-Practice community is the blueprint for how to run, govern, and sustain such a routine.

References

1. Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall PTR.
2. Duarte, V. (2016). *#NoEstimates: How to Measure Project Progress Without Estimating*. Oikosofy Series.
3. Magennis, T. (2011). *Forecasting and Simulating Software Development Projects: Effective Methods for Realistic Predictions*. Focused Objective.
4. Magennis, T. (2016). *When Will It Be Done? Lean-Agile Forecasting to Answer Your Customers' Most Important Question*. Focused Objective.
5. Vacanti, D. S. (2015). *Actionable Agile Metrics for Predictability: An Introduction*. ActionableAgile Press.
6. Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide*. Scrum.org.
7. Jeffries, R. (2019). Issues with Story Points. RonJeffries.com Blog. Retrieved from https://ronjeffries.com/articles/019-01ff/story-points/Index.html
8. Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.
9. Harrell, F. E. (2015). *Regression Modeling Strategies* (2nd ed.). Springer.
10. Hastie, T., Tibshirani, R., & Friedman, R. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.
11. Efron, B., & Tibshirani, R. J. (1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC.
12. Boehm, B. W. (1981). *Software Engineering Economics*. Prentice-Hall.
13. Jørgensen, M., & Shepperd, M. (2007). A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, 33(1), 33–53.

Appendix A. Minimal model transparency

This appendix provides a minimal throughput functional form for transparency. The manuscript's primary contribution is the operational routine and governance; practitioners can implement the simulation with simpler assumptions if desired.

The implementation modeled a daily throughput rate from historical signals using a simple linear form. Model parameters were estimated from the rolling historical window using ordinary least squares and then simulated future weeks under uncertainty:

$$\text{daily\_rate}_t = \beta_0 + \beta_1 \cdot \text{prev\_daily\_rate}_{t-1} + \beta_2 \cdot \text{unplanned\_fraction}_t + \beta_3 \cdot \text{percent\_bug}_t + \varepsilon_t \quad \text{(A1)}$$

In simulation, we propagate uncertainty by (i) sampling β from a multivariate normal distribution defined by the fitted covariance matrix (when available) and (ii) sampling ε from N(0, σ). For each simulated sprint we construct x by sampling a historical feature row from the rolling window (with replacement), and we truncate simulated throughput at zero to avoid negative productivity draws.

In practice, the specific functional form is less important than (i) producing a stable percentile signal under uncertainty, and (ii) institutionalizing how that signal is used: commitments anchored to P90; and corrective actions triggered by widening uncertainty or drifting percentiles.

Data & Code Availability: Synthetic dataset generator, source code, and a sample anonymized report artifact: https://github.com/berkkibarer/agile_release_forecast | Archived version with DOI: https://doi.org/10.5281/zenodo.18361692