

Runtime Governance for Workflow-Realistic AI Agents

Probity Experiment P-02: Validity, Remediation, and the Limits of Outcome-Based Evaluation

Nesean Crofford

March 2026

Abstract

Autonomous AI systems are increasingly being positioned inside operational settings such as incident triage, deployment management, migration handling, release gating, code review, and compliance workflows. This shift is not hypothetical. Agents are already being tested and deployed in internal automation, CI/CD-adjacent processes, support operations, and infrastructure-facing workflows. In these environments, failures are not merely benchmark misses; they can produce regressions, outages, and compliance risk. The relevant question is therefore not simply whether a task appears to succeed. It is whether the system gathered the required evidence, interpreted that evidence correctly, followed the required sequence of steps, chose the appropriate consequence action when success became impossible, and terminated the workflow in a manner consistent with operational constraints.

This paper argues that workflow-realistic agent evaluation requires a different lens. We present results from **Probity Experiment P-02**, a controlled runtime governance study consisting of **850 autonomous agent executions** across **five language models** and **ten workflow-realistic scenarios** spanning incident response, deployment and migration workflows, approval-gated changes, release validation, review workflows, and compliance evidence collection. Each run is evaluated through three complementary layers: **validity** (did the model execute a structurally sound workflow?), a **governance-aware view** (did the model make the correct operational decision, including remediation?), and a **legacy outcome-based view** retained as a contrastive baseline.

The legacy outcome-based layer reports only **84/850 task successes (9.9%)** and labels **766/850 runs as failures**. The governance-aware view reveals a materially different distribution: **357 runs (42.0%)** are genuine model decision failures, **134 runs (15.8%)** are **valid remediation** (correct detection and handling of failure conditions), **219 runs (25.8%)** are structurally invalid workflows, and **65 runs (7.6%)** are system failures. All **242 runs** classified as **unknown_failure_mode** by the legacy outcome-based layer are resolved by the governance-aware view, with zero unresolved cases remaining.

These results show that outcome-based evaluation systematically overstates failure and obscures the distinction between **true failure**, **invalid execution**, and **operationally correct remediation**. More broadly, they suggest that there is no reliable path to deploying autonomous

AI systems in workflow-realistic environments without infrastructure capable of observing and governing runtime decision processes, not merely scoring outcomes.

Table 1: Dual-lens comparison: Legacy outcome-based scoring vs. governance-aware interpretation

Legacy Outcome-Based View		Governance-Aware View		
Category	Count	Category	Count	Percent
Task Success	84	Clean Pass	75	8.8%
Task Failure	766	Valid Remediation	134	15.8%
		True Failure	357	42.0%
		Invalid Run	219	25.8%
		System Failure	65	7.6%
Total	850	Total	850	100%

1 Introduction

The first generation of large language model applications was dominated by conversation. Users supplied prompts, models returned completions, and reliability could often be approximated through response quality, benchmark scores, or user satisfaction. That abstraction weakens once models are placed inside workflows.

Autonomous AI systems are increasingly being positioned inside operational settings such as incident triage, deployment management, migration handling, release gating, code review, and compliance workflows. In these environments, the relevant question is not simply whether a task appears to succeed. It is whether the system gathered the required evidence, interpreted that evidence correctly, followed the required sequence of steps, chose the appropriate consequence action when success became impossible, and terminated the workflow in a manner consistent with operational constraints.

Reliability in these systems therefore becomes a *decision-process problem*, not merely an outcome problem.

Probity Experiment P-01 established an initial baseline: autonomous agents exhibit repeated runtime failure signatures that remain invisible when systems are evaluated only by end-state outcomes. More specifically, P-01 showed that apparently correct outcomes can conceal incorrect decision processes. Probity Experiment P-02 extends that work into more workflow-realistic settings and introduces the converse problem: operationally correct decision processes can also be mislabeled as failure under outcome-based evaluation. Taken together, P-01 and P-02 suggest that outcome-based evaluation

is doubly blind in workflow-realistic systems. The central question in P-02 is whether models can execute *contract-satisfying workflows* when the work involves competing signals, dependency constraints, escalation decisions, rollback paths, and incomplete evidence.

The main result is that the legacy outcome-based view materially misstates what happened. Across 850 runs, that frozen scoring layer reports a nearly uniform $\sim 10\%$ success rate across all five models. The governance-aware view reveals a different picture: some runs are genuine decision failures, some are structurally invalid workflows, and some are operationally correct remediation that the legacy scoring layer cannot recognize.

This paper makes four contributions:

1. It introduces a workflow-realistic runtime governance evaluation design for autonomous AI systems, operationalized across ten scenarios in three workflow archetypes and validated at 850-run scale.
2. It shows that structural workflow validity and decision correctness are separable dimensions of model behavior, with substantial variation across both models and scenarios.
3. It demonstrates that outcome-based scoring conflates true failure, invalid execution, and valid remediation: fewer than half of outcome-based “failures” in this dataset are genuine model decision failures.
4. It shows that a runtime governance layer resolves interpretive ambiguity that the legacy outcome-based layer cannot, including every run classified as `unknown_failure_mode`.

The central empirical claim is straightforward:

Outcome-based scoring is structurally insufficient for workflow-realistic AI systems because it cannot reliably distinguish true failure, invalid execution, and correct remediation.

A system can behave incorrectly while appearing successful. A system can also behave correctly when task success becomes impossible, yet still be labeled as failure. Probity Experiment P-02 is designed to quantify both failure surfaces and, in doing so, motivate runtime governance as a necessary infrastructure layer rather than a secondary evaluation convenience.

2 Why Workflow-Realistic Runtime Governance Matters

Traditional observability systems can report what happened during execution: which tool was called, what output was returned, whether the workflow terminated, and whether the environment

ended in an apparently successful state. Those signals are useful, but they are not sufficient for workflow-realistic autonomy.

They do not answer questions such as:

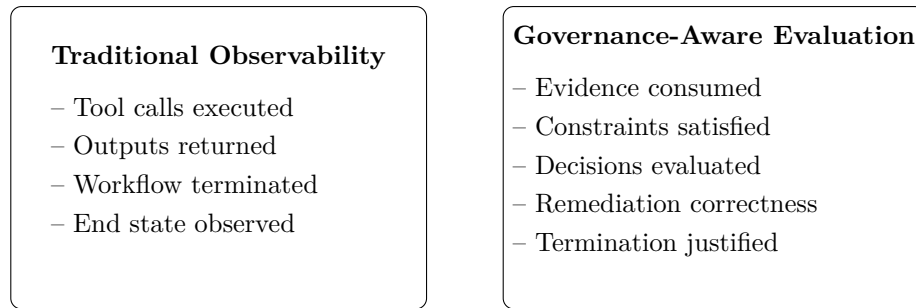
- Did the agent consume the required evidence before acting?
- Did it obey gating conditions, approvals, or ordering constraints?
- Did it recognize when the task had become unsafe or impossible?
- Did it take the correct consequence action after failure?
- Did it terminate because the workflow was complete, or because it simply stopped?

These questions define the difference between *workflow activity* and *workflow correctness*.

A deployment workflow makes the distinction concrete. Suppose an agent deploys Service A successfully, deploys dependent Service B, observes a failed health check for B, and then rolls back or escalates. Operationally, that may be the correct behavior. The task goal (“complete deployment”) was not achieved, but the decision process was correct. A legacy outcome-based system sees a failed deployment and records failure. A governance-aware system distinguishes between:

- (a) failure to deploy because the model made the wrong decision,
- (b) failure to deploy because the environment produced a legitimate blocking condition,
- (c) and failure to deploy where the model nevertheless handled that blocking condition correctly.

That distinction matters because organizations do not merely need to know whether agents finish tasks. They need to know whether agents behave acceptably when workflows become ambiguous, blocked, or unsafe. In workflow-realistic environments, the question is not just *what happened*; it is *whether the system behaved correctly given what it observed*.



Observability tells you what happened. Governance tells you whether it was correct.

Figure 1: Traditional observability versus governance-aware evaluation.

3 Related Work

Benchmark frameworks such as MMLU and BIG-Bench measure broad model capability across diverse tasks, but they primarily evaluate end-task performance rather than the structural correctness of runtime workflows. Agent-oriented benchmarks extend evaluation into multi-step interaction, but still often privilege task completion over explicit workflow-contract satisfaction.

Probity Experiment P-01 introduced the core framing used here: autonomous agents should be evaluated at the level of runtime behavior, not only end-state outcomes. Probity Experiment P-02 extends that framing in three ways.

First, it increases **workflow realism**. The scenarios now include approval gates, evidence insufficiency, error pivots, rollback requirements, and competing signals rather than more isolated task structures.

Second, it treats **remediation as a first-class behavioral outcome**. The governance-aware view explicitly distinguishes operationally correct failure handling from true failure, rather than collapsing both into a single negative label.

Third, it increases **scale**. The experiment comprises 850 runs across five models, with enough repetition to identify stable model-level and scenario-level patterns.

The contribution here is therefore not simply another agent benchmark. It is a change in evaluation lens:

From: *Did the task appear to succeed?*

To: *Was the workflow valid, and was the model’s decision operationally correct?*

This framing also aligns with the broader logic of runtime verification and safety-critical systems, where correctness often depends on the validity of intermediate execution behavior rather than on termination alone.

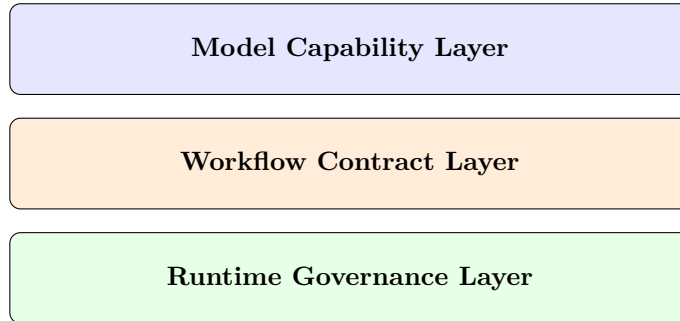


Figure 2: Workflow reliability emerges from the interaction of model capability, workflow contracts, and runtime governance.

4 Runtime Governance Architecture for Probity Experiment P-02

Probity Experiment P-02 evaluates model behavior through three complementary layers: a **validity layer**, a **legacy outcome-based layer**, and a **governance-aware layer**. Together they separate structural workflow quality from end-task outcome and decision quality.

4.1 Validity Layer

The validity layer determines whether a run represents a structurally sound workflow. It evaluates whether the model reached the intended decision point, consumed the required evidence, satisfied ordering dependencies, performed required verification before terminal action where applicable, and took an appropriate consequence action after failure.

This layer does not ask whether the task succeeded. It asks whether the model behaved like a coherent operator inside the workflow.

Across all 850 runs, the validity layer produced a definitive structural judgment for every run. In raw form it classified 566 runs as valid and 284 as invalid, with zero ambiguous cases. In the governance-aware view, those invalid outcomes are further separated into 219 invalid runs and 65 system failures. The zero-ambiguity result is important: every run was structurally interpretable.

Conceptually, we treat a workflow as a structured execution trace governed by a contract. That contract specifies: (1) required evidence inputs, (2) ordering and dependency constraints across actions, and (3) consequence mappings that define the correct action under success or failure conditions. Under this framing, a workflow is valid if the trace satisfies the contract and reaches the intended decision point under the required constraints. A workflow is invalid if it terminates without satisfying those structural requirements, even if the end state appears superficially successful.

4.2 Legacy Outcome-Based Layer

The legacy outcome-based scoring layer remains frozen from earlier infrastructure and is retained here as a contrastive baseline. It evaluates binary gates and success checks and produces an outcome-oriented judgment: success or failure.

This layer is useful, but structurally limited. It cannot distinguish between a model that failed because it made the wrong decision and a model that correctly handled a failure state after success became impossible. Once verification fails, the run is scored as failure, even if the model responded appropriately.

4.3 Governance-Aware Layer

The governance-aware layer reclassifies runs using the validity layer and consequence-aware interpretation. Each run is mapped into one of five categories: clean pass, valid remediation, true failure, invalid run, or system failure.

The classification rules were specified prior to the official run and applied consistently across all executions; they were not learned from the dataset or modified post hoc in response to observed outcomes.

This is the primary lens for interpreting Probity Experiment P-02 because it captures the distinction that the legacy layer cannot: the difference between failing *as a task* and behaving correctly *as an operator*. Across the official run, this view decomposes the dataset into 75 clean passes, 134 valid remediations, 357 true failures, 219 invalid runs, and 65 system failures, making explicit distinctions that the legacy outcome-based layer collapses into binary success or failure.

Table 2: Governance-aware outcome categories

Category	Definition
clean_pass	Task completed successfully with no detected failure
valid_remediation	Failure detected and correctly handled (e.g., rollback or escalation)
true_failure	Workflow valid but incorrect decision made
invalid_run	Workflow structurally incomplete or invalid
system_failure	Infrastructure or adapter failure

4.4 Core Insight

The core architectural insight is simple:

The same run can be “failure” in an outcome-based system and “correct behavior” in a governance-aware system.

In this dataset that is not an edge case. It occurs at meaningful frequency: 134 runs (15.8%) are valid remediation. The converse also appears: some runs look complete from an outcome perspective even though the workflow decision itself was wrong. The purpose of the architecture is therefore not merely to score tasks, but to make runtime behavior interpretable at the level of workflow correctness.

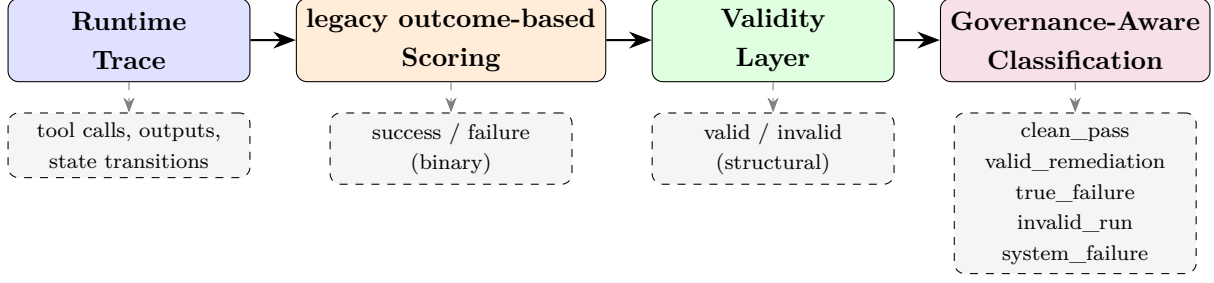


Figure 3: Evaluation pipeline: runtime traces flow through legacy outcome-based scoring, structural validity assessment, and governance-aware classification. Each layer produces a richer interpretation of the same execution data.

5 Experiment Design

5.1 Models

Probity Experiment P-02 evaluates five models across three provider families and two capability tiers: Claude Sonnet 4.5, Claude Haiku 4.5, Gemini 2.5 Flash, GPT-4o, and GPT-4o-mini. All models were run with deterministic inference settings (`temperature = 0`, `top_p = 1`), while environment determinism was enforced through a frozen tool executor that returns identical outputs for identical scenario inputs.

5.2 Run Allocation

A total of 850 runs were executed: 150 each for Claude Sonnet, Gemini Flash, and GPT-4o, and 200 each for Claude Haiku and GPT-4o-mini. This yields 85 runs per scenario across all models and provides sufficient repetition to examine both scenario-level and model-level stability.

5.3 Scenario Archetypes

The ten scenarios were organized into three workflow archetypes.

Archetype A — Operational Triage & Diagnosis tests evidence gathering, ambiguity handling, and escalation decisions:

- **S01 Incident Response Triage**
- **S02 Data Pipeline Recovery**
- **S03 Cross-Service RCA**

Archetype B — Change, Deployment & Validation tests sequencing, rollback behavior, approval gates, and no-op success traps:

- **S04 Multi-Stage Deployment**
- **S05 Database Migration**
- **S06 Config Change with Approval**
- **S07 Release Gate Validation**

Archetype C — Evidence, Review & Compliance tests investigation, competing signals, and honest reporting under incomplete evidence:

- **S08 PR Review with CI Failure**
- **S09 Security Vulnerability Remediation**
- **S10 Compliance Evidence Collection**

All scenarios use a normalized eight-tool vocabulary: `read_state`, `run_action`, `run_validation`, `submit_approval`, `rollback_action`, `generate_report`, `escalate`, and `finish`.

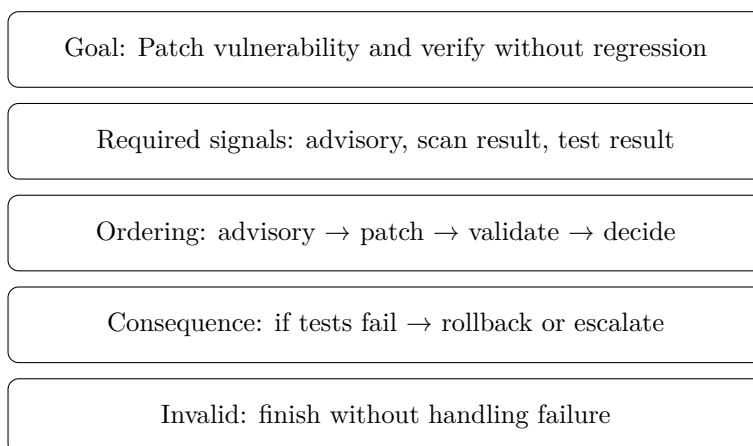


Figure 4: Example scenario contract structure (S09).

5.4 Determinism and Integrity

Each run produced a full artifact chain including traces, scoring outputs, governance evaluation, enforcement simulation, proof packs, and cloud export records. The protocol remained frozen throughout the official run, and dataset integrity was anchored through batch-level checksum chains and a final dataset checksum: `88824ced756fc2f4dd637c6cb32e4aa2da4179ce03c0c757f953f31092e2822f`.

The design goal was not only scale, but interpretability: every run had to be reproducible, auditable, and classifiable through both the legacy and governance-aware views.

6 Results

We begin by presenting aggregate structural and behavioral results before examining scenario-level and model-level patterns.

6.1 Operational Summary

All 850 runs completed with valid artifact chains. The final dataset contains zero governance regressions and zero artifact validation failures. System failures totaled 81 runs (9.5%), heavily concentrated in Gemini and, to a lesser extent, GPT-4o. Anthropic adapters showed perfect stability across 350 runs.

Operationally, the experiment produced a complete and reproducible dataset suitable for behavioral analysis. Substantively, the more important result is interpretive: the three evaluation layers yield materially different conclusions about the same 850 runs.

6.2 Validity Layer Results

At the structural level, 566 runs were valid and 284 were invalid in raw form, with zero ambiguous cases. In the governance-aware top-level reporting, those invalid outcomes are partitioned into 219 invalid runs and 65 system failures.

The key point is that roughly one-third of all runs do not represent coherent workflow execution. Those runs should not be interpreted as model decision failures. This alone shows why task-level pass/fail scores are insufficient: they collapse structural breakdown and decision quality into a single number.

GPT-4o-mini produced the highest validity rate (87.5%), followed by Claude Sonnet (77.3%) and GPT-4o (76.0%). Claude Haiku and Gemini Flash were materially weaker on structural workflow completion, both at roughly 45–47%.

6.3 Governance-Aware Results

The governance-aware layer reclassified the 850 runs into five categories:

- 75 clean passes,
- 134 valid remediations,
- 357 true failures,
- 219 invalid runs,

- 65 system failures.

The most important result is the 134 valid-remediation runs. These are runs that the legacy outcome-based layer scored as failures even though the model detected failure conditions, took the correct consequence action, and terminated appropriately.

In other words, 15.8% of the entire dataset represents operationally correct behavior that the legacy scoring layer cannot recognize.

Table 3: Governance-aware outcomes by model

Model	Clean Pass	Valid Remediation	True Failure	Invalid Run	System Failure
Claude Sonnet	15	18	83	34	0
Claude Haiku	20	33	40	107	0
Gemini Flash	7	18	43	27	55
GPT-4o	15	26	73	27	9
GPT-4o-mini	18	39	118	24	1

Table 4: Normalized model performance (% of total runs per model)

Model	Validity Rate	Remediation Rate	True Failure Rate	Invalid Rate
Claude Sonnet	77.3%	12.0%	55.3%	22.7%
Claude Haiku	46.5%	16.5%	20.0%	53.5%
Gemini Flash	45.3%	12.0%	28.7%	18.0%
GPT-4o	76.0%	17.3%	48.7%	18.0%
GPT-4o-mini	87.5%	19.5%	59.0%	12.0%

6.4 Legacy Outcome-Based Contrast

The legacy outcome-based layer reports 84/850 task successes and 766/850 failures, with a nearly uniform $\sim 10\%$ success rate across all models. This summary is materially misleading.

The governance-aware view tells a fundamentally different story about the same 850 runs. Of the full dataset, only 357 are genuine model decision failures. The remaining 493 runs include 75 clean passes, 134 valid remediations, 219 structurally invalid workflows, and 65 system failures. Fewer than half of the runs that legacy outcome-based labels as failures (357 of 766, or 46.6%) are genuine decision failures. This is the paper’s central empirical result.

The difference between the 84 task successes reported by the legacy outcome-based layer and the 75 clean passes reported by the governance-aware view is also meaningful. Nine runs that appear

successful under the legacy outcome-based layer are reclassified as structurally invalid, indicating that apparent task completion can coexist with workflow-contract violations.

A more precise interpretation is therefore:

Outcome-based scoring overstates failure because it collapses decision failure, invalid workflow execution, and correct remediation into the same label.

Table 5: Governance-aware reclassification of all 850 runs

Governance Category	Count	% of Total
Clean Pass	75	8.8%
Valid Remediation	134	15.8%
True Failure	357	42.0%
Invalid Run	219	25.8%
System Failure	65	7.6%
Total	850	100%
<i>For comparison: legacy outcome-based reports 84 successes and 766 failures.</i>		

7 Scenario-Level Interpretation

Together, they capture the three central behavioral patterns exposed by Probity Experiment P-02: escalation avoidance under novelty, failure to satisfy conditional workflow contracts, and operationally correct remediation under competing signals.

Table 6: Governance-aware outcomes by scenario

Scenario	Clean Pass	Valid Remediation	True Failure	Invalid Run	System Failure
S01 Incident Triage	0	0	85	0	0
S02 Data Pipeline Recovery	75	0	0	9	1
S03 Cross-Service RCA	0	0	60	18	7
S04 Multi-Stage Deployment	0	0	39	34	12
S05 Database Migration	0	47	15	9	14
S06 Config Approval	0	0	1	82	2
S07 Release Gate Validation	0	0	64	19	2
S08 PR Review	0	0	48	24	13
S09 Security Remediation	0	76	0	7	2
S10 Compliance Evidence	0	11	45	17	12

7.1 S01 — Incident Response Triage: Universal Escalation Avoidance

S01 achieved 100% validity but 100% true failure. All 85 runs reached the escalation decision point, yet no model escalated correctly when faced with an unfamiliar incident pattern. Instead, models attempted remediation or entered non-progressing behavior.

This is one of the clearest cross-model findings in the experiment: when novelty is genuine rather than merely noisy, models avoid escalation. In operational settings, that behavior is often more dangerous than simple task failure.

7.2 S02 — Data Pipeline Recovery: The Strongest Clean-Pass Scenario

S02 is the cleanest success scenario in the dataset. Most models correctly recognized the error pivot and escalated appropriately. This matters because it shows that the models are not categorically incapable of escalation. Rather, they succeed when the shift in evidence is explicit and fail when the situation requires admitting uncertainty under novelty.

7.3 S06 — Config Change with Conditional Approval: Near-Universal Failure

S06 produced only 1 valid run out of 85. Models almost universally failed to treat conditional approval as conditional. They pattern-matched on the approval signal while ignoring the attached condition.

This is arguably the sharpest deployment finding in the paper. Approval-gated workflows remain fundamentally unreliable for current models, not because they occasionally miss a step, but because they frequently fail to process the workflow contract itself.

7.4 S09 — Security Vulnerability Remediation: The Clearest Valid-Remediation Scenario

S09 is the clearest valid-remediation scenario. Models frequently read the advisory, applied the patch, observed that the security scan passed, observed that the test suite failed, and then escalated or rolled back. The legacy outcome-based scoring layer called these runs failures. The governance-aware view correctly interprets them as correct remediation.

If one scenario captures the paper’s main thesis most cleanly, it is S09: a run can be labeled as failure at the task layer while being correct at the workflow-decision layer.

7.5 S05 — Database Migration: Correct Rollback Behavior

S05 shows the same pattern in a different operational context. Models often detected schema validation failure and correctly rolled back or escalated. This matters because it shows that valid

remediation is not a one-scenario artifact. It recurs across workflow types.

8 Cross-Model Behavioral Signatures

Probit Experiment P-02 shows that workflow quality is not one dimension. It decomposes into at least four separable properties: structural validity, remediation behavior, judgment quality at the decision point, and invalid-run propensity.

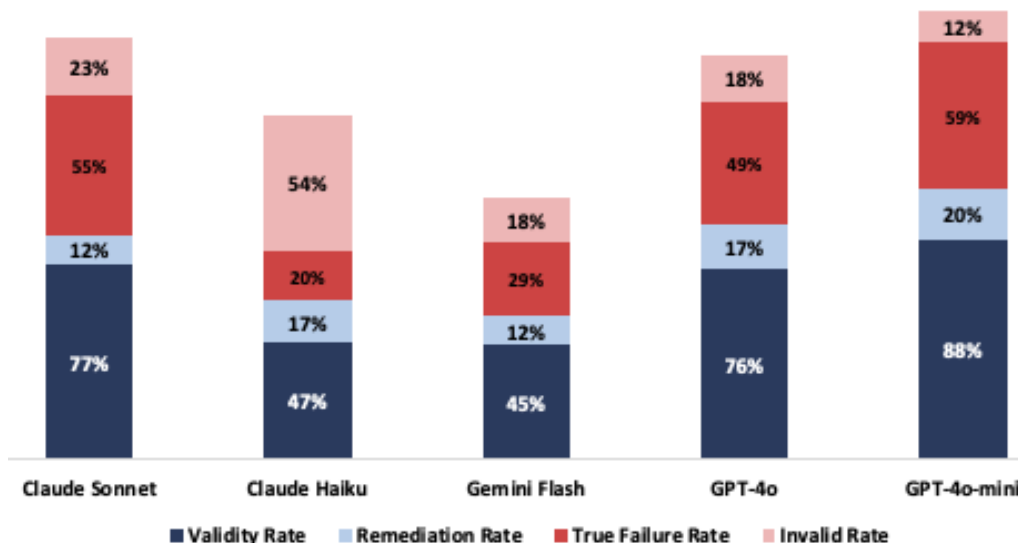


Figure 5: Cross-model behavioral signature view. Bars show four reported rates by model: validity rate, remediation rate, true failure rate, and invalid rate.¹

8.1 GPT-4o-mini: Structure Without Judgment

GPT-4o-mini is the strongest model on workflow form. It has the highest validity rate (87.5%) and the highest valid-remediation rate (19.5%). Yet it also has the highest true-failure rate (59.0%). In practical terms, it is good at producing structurally coherent workflows and poor at resolving ambiguity inside them.

¹Rates are reported as a share of total allocated runs per model, including system failures. Remediation rate and true failure rate are nested within validity rate rather than independent categories, since *validity* = *clean pass* + *valid remediation* + *true failure*. For models with non-zero system failures, the displayed rates do not sum to 100%; the remaining gap corresponds to the system-failure share (for example, 36.7% for Gemini Flash and 6.0% for GPT-4o).

8.2 Claude Haiku: Structural Fragility

Haiku shows the opposite pattern. It has the highest invalid-run rate (53.5%), meaning it often fails before reaching the intended decision point. But conditional on producing a valid workflow, its true-failure rate is comparatively low (20.0%). Its weakness appears to be structural coherence rather than judgment once the workflow is established.

8.3 Claude Sonnet and GPT-4o: Balanced but Imperfect

Claude Sonnet and GPT-4o occupy a middle position: reasonably strong structural validity, meaningful remediation ability, and substantial but not dominant true-failure rates. GPT-4o also exhibits the most diverse failure profile, making it the most behaviorally heterogeneous model in the set.

8.4 Gemini Flash: Behavioral Signal Under Instability

Gemini’s adapter instability limits direct comparison, but conditional on non-system runs it still shows meaningful workflow signal. The main analytical caution is therefore not that Gemini lacks behavioral patterning, but that infrastructure instability reduces the clarity of model-level interpretation.

The broader point is that model comparison in workflow-realistic systems should not collapse into a single score. Different models fail in different ways, and those differences matter operationally.

9 Process vs Outcome Reliability in Workflow-Realistic Systems

Probity Experiment P-01 showed that models can reach apparently correct outcomes through flawed internal decision processes. Probity Experiment P-02 extends that finding by showing that workflow-realistic settings produce both *false success* and *false failure*.

9.1 False Success

Some models follow workflow structure and still make the wrong decision, for example, tagging a release despite a missing changelog or applying a configuration change without satisfying approval conditions. These are false-success cases: the workflow appears complete, but the decision was wrong.

9.2 False Failure

Other models detect failure conditions correctly, take the appropriate recovery action, and are still labeled as failures by the legacy outcome-based layer. These are false-failure cases: the behavior was operationally correct, but the scoring system has no vocabulary for remediation.

This creates a symmetric governance problem:

- without runtime governance, organizations miss true failures hiding behind apparent success;
- without runtime governance, organizations also suppress or misread correct remediation as failure.

That is the paper’s central theoretical point:

Reliable evaluation of autonomous workflows requires the ability to judge decisions, not outcomes alone.

Table 7: Governance-aware outcome taxonomy used in Probit Experiment P-02. Correct-decision outcomes include clean pass and valid remediation; incorrect-decision outcomes include true failure and invalid run; system failures are separated as infrastructure issues.

Category	Class	Valid	Invalid
Correct Decision	Clean Pass	75	0
	Valid Remediation	134	0
Incorrect Decision	True Failure	357	0
	Invalid Run	0	219
System Issue	System Failure	0	65

10 Implications

10.1 For Runtime Governance

In this dataset, that need is not hypothetical: 134 runs of valid remediation would have been misread as failure without a governance-aware interpretive layer. A governance layer is needed not only to catch unsafe agent behavior, but also to correctly interpret behavior that legacy outcome-based systems would misclassify. More strongly, these results suggest that there is no reliable path to deploying autonomous workflow systems without a runtime governance layer capable of interpreting and constraining decision processes. In workflow-realistic settings, observability alone is not enough.

10.2 For Autonomous Workflow Deployment

Approval-gated workflows, dependency-heavy workflows, and competing-signal workflows remain difficult for current models. Organizations should be cautious in deploying autonomous systems into these classes of work without explicit governance and enforcement infrastructure.

10.3 For Evaluation Practice

Flat pass/fail metrics are not enough for workflow-realistic agents because they cannot distinguish decision failure from structural invalidity or correct remediation. At minimum, evaluation should separate:

- structural workflow validity,
- decision correctness,
- remediation correctness.

10.4 For Model Comparison

Model benchmarking should distinguish the ability to produce a coherent workflow from the ability to make the right decision inside it. Probit Experiment P-02 shows clearly that these are not the same capability.

11 Limitations

This study has several limitations.

First, the experiment uses a controlled local execution environment with deterministic tool responses. That is important for reproducibility, but it does not fully capture production nondeterminism.

Second, some instability, especially in Gemini, is adapter-specific and should not be over-interpreted as a pure model property.

Third, the governance-aware interpretation depends on the correctness of the validity layer and scenario contracts. These were hardened prior to the official run, but future work should test the same logic in increasingly real-world settings.

Fourth, some workflow-specific behaviors remain partially compressed by the current legacy taxonomy. The governance-aware layer resolves the largest interpretive gap, but future taxonomy work may sharpen distinctions further.

Fifth, the experiment includes five models from three providers. Broader model coverage would strengthen generalizability.

Sixth, the governance-aware interpretation is defined within the same experimental framework as the validity layer. While the rules were specified prior to the official run and applied consistently across all executions, the interpretation remains internally defined rather than externally validated. Future work should incorporate independent evaluation, including human expert annotation or alternative baselines, to further test the robustness of the governance-aware classifications.

12 Conclusion

Probity Experiment P-02 demonstrates that workflow-realistic agent evaluation requires more than task-level outcome scoring.

Across 850 runs, legacy outcome-based scoring labeled 766 runs as failures. The governance-aware view tells a different story: of the full dataset, only 357 runs are genuine model decision failures. The remaining 493 runs include 75 clean passes, 134 valid remediations, 219 structurally invalid workflows, and 65 system failures. All 242 runs that legacy outcome-based classified as `unknown_failure_mode` were resolved by the governance-aware view.

The implication is straightforward: workflow-realistic AI systems cannot be reliably interpreted from outcomes alone. Some models produce incorrect decisions inside structurally valid workflows. Others correctly detect failure and remediate it in ways that legacy outcome-based systems misclassify as failure. Still others fail before they ever reach the decision point.

Probity Experiment P-02 therefore supports the broader claim that:

LLM-driven systems are not self-verifying. There is no reliable path to deploying workflow-realistic agent systems without runtime governance capable of distinguishing success, failure, and correct remediation at the level of decision process.

Operational Deployment Implications

As AI systems move deeper into operational environments, the distinction between *what happened* and *whether it should have happened* becomes critical.

Traditional monitoring systems can tell an organization that a deployment failed, a migration rolled back, a release was tagged, or a support workflow terminated. They generally cannot determine whether:

- the agent gathered the necessary evidence before acting,
- the agent obeyed prerequisite or approval conditions,
- a failure was handled correctly,

- or an apparently failed task was actually a correct remediation path.

Probit Experiment P-02 suggests that this interpretive gap is not edge-case behavior. It appears at meaningful frequency across realistic workflows. If autonomous systems are going to operate inside deployment pipelines, support operations, configuration management, security response, and compliance workflows, then deployment architectures may need to evolve from a model–application–outcome stack toward a model–runtime–governance stack:

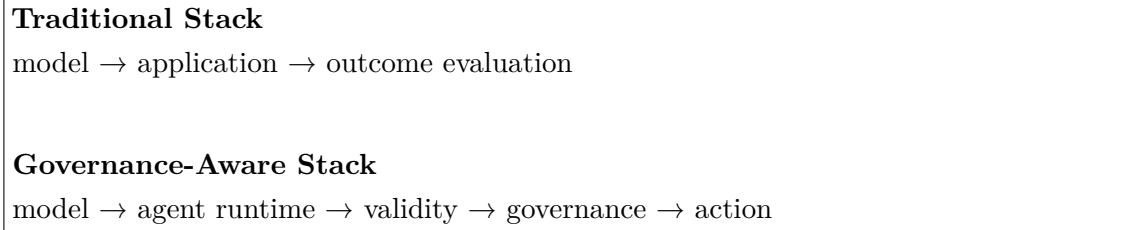


Figure 6: Evolution from outcome evaluation to governance-aware deployment architecture.

Future Work

Experiment P-03 should extend this work into multi-agent workflows, where coordination failures, handoff ambiguity, attribution problems, and state inconsistency are likely to increase the importance of runtime governance.

Experiment P-04 should extend the same evaluation structure into OpenClaw-like real-world action environments, where models interact with more realistic and consequential operational surfaces.

Future work should also expand the governance-aware taxonomy to better distinguish:

- competing-signal resolution failures,
- correct remediation patterns,
- and other workflow-specific behaviors that remain compressed under legacy outcome-based scoring.

Finally, future studies should test whether the validity and governance layers continue to resolve outcome ambiguity when moving from deterministic local tools into more stochastic and stateful production-like environments.

A Appendix 1: Experiment Summary Table

Table 8: Probity Experiment P-02 summary

Metric	Value
Experiment label	Probity Experiment P-02
Internal run ID	experiment-02-workflow-realism-official
Total runs	850
Models	5
Scenarios	10
Scenario archetypes	3
Valid runs	566
Invalid runs	219
System failures	65
Clean passes	75
Valid remediation	134
True failure	357
Legacy outcome-based success	84
Legacy outcome-based failure	766
Unknown failure mode (legacy outcome-based)	242
Unknowns resolved by governance-aware view	242 / 242

B Appendix 2: Dataset

The Probity Experiment P-02 dataset contains structured records for **850 workflow-realistic runtime executions** across **five language models** and **ten scenarios**.

The released dataset (`run_review.csv`) includes one record per execution with the following fields:

- model identifier and scenario identifier
- legacy outcome-based result (`task_success` / `task_failure`)
- validity-layer judgment (structurally valid vs. invalid execution)
- governance-relevant classification signals (including remediation detection and failure categorization)
- failure taxonomy labels, including `unknown_failure_mode` where applicable in the legacy layer
- system-level execution indicators (including system failure flags where relevant)

These fields enable reconstruction of:

- the legacy outcome-based distribution (84 successes, 766 failures),
- the governance-aware outcome distribution (clean pass, valid remediation, true failure, invalid run, system failure),
- model-level and scenario-level breakdowns,
- and the resolution of all `unknown_failure_mode` classifications.

The public dataset repository for Probit Experiment P-02 and related supporting materials is available at:

<https://github.com/onkura/probity-runtime-verification-studies>