Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Failure Taxonomy of an Agent-Run Research Lab: 105 Documented Process Failures, Who Caught Them, and an Open Dataset

Kiyoshi Casey

kiyoshi@321ai.xyz

Florida State University https://orcid.org/0009-0006-6256-9325

Research Article

Keywords: ai-agents, research-operations, failure-taxonomy, dataset, governance, llm

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11662926/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Failure Taxonomy of an Agent-Run Research Lab: 105 Documented Process Failures, Who Caught Them, and an Open Dataset

Kiyoshi Casey - 321 AI Labs - ORCID 0009-0006-6256-9325

2026-11-05

Edition: TechRxiv submission edition (adds bounded related work, declarations, references).
Edition source sha256: 8f9f600bfaf8bc00c3e27299b4320f30ad7057675a2eca7c631b01635e38de4a
Derived from frozen pub copy glm/pub/p5/TR5-SKELETON.md sha256
cdf70c98d1fe41f649c1535028777cb8342cb4aacb187a6e8b829e1da5f15af4
Frozen work baseline: glm/work/p5/report/TR5-SKELETON.md sha8 cef52b53
Dataset: glm/work/p5/failure-dataset/records.jsonl v1.1 sha256
6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee
W2 canaries embedded (IDs only; registry private): W2-C-04, W2-C-05
Rendering note: fleet-internal envelope JSON, STATUS line, figure/table map, and report notes are omitted from this PDF; the canonical artifact is the frozen markdown.

Abstract

We present a failure taxonomy built from a research fleet's own operational record: N = 105 documented failure records drawn from four source families of one agent-run lab (flashnext 33, sprint-review 60, timesfm 7, job-killer 5), September 11–15 2026 [dataset-metadata.json, 7d2fb896143cf582769978caae92114a1335e4df6efda2398bf9388a0c2422db]. Each record classifies one documented failure along three controlled vocabularies: where in the pipeline it lived (phase), why it happened (cause_class), and which governance layer caught it (detection_layer). Each record also pins a verifiable citation to the failure's documentation.

Three results stand out. First, failures concentrate in the verify/build phases (build 35, verify 27, ops 25, spec 18) and in one cause class: verification_gap accounts for 46 of 105 records (44%): checks that existed but were silently never exercised dominate the corpus, ahead of environment_drift (17) and spec_drift (16) [dataset-metadata.json, 7d2fb896…; Fig1]. Second, the fleet's layering caught everything that was documented: detection_layer resolves to self_caught 72/105 (69%) and reviewer_caught 33/105 (31%), with owner_caught 0 and escaped 0, reported as measured [dataset-metadata.json, 7d2fb896…; EXTRACTION-REPORT.md, c303fdf17813905d43349fbe3be510eba97b54a02d8aad05a98b1e6953788d59], stated throughout with the dataset card's survivorship/definition caveat, quoted verbatim in §7, never softened. Third, the classification itself is audited: two independent blind classifiers labeled a 40-record stratified sample, and per-field Cohen's κ rose across two logged taxonomy revisions from 0.728/0.458/0.946 to 0.965/1.000/1.000 (cause_class/phase/detection_layer). The audit trail, including a contested-record adjudication and sealed label hashes, is itself a method contribution of this report [kappa-audit-trail.json, 3be3969bf441af9e9d5cc09a37135b69d00e036da52894731b0460eff0dfc369].

The dataset is versioned (v1.1), deterministic (run-twice hash-compare with an externally anchored digest), machine-readable, and published with an open submission format for external contributions; it directly feeds the fleet's W1 recursive-governance mission, whose amendment proposals must cite these records [README.md card, 9219fd107ea0e9ef534edb0b4b228098656626e92dd8c3bd5983945cb1ca807f; SUBMISSION-FORMAT.md, 0c2b3fe184dfc0d584c34842ae5e4b645ac0948db7f87af7330efb18ed5e30d5]. In keeping with the governance contract's null/negative rule we also report what the corpus cannot support: cost coverage is 3 measured / 1 estimate / 101 null, and no cost distribution is reported because 101 of 105 sources state no cost. Any future cost figure requires a pre-defined estimate protocol with every derived value labeled `estimate` (ruling: BRIEF-20260915-p5-w1-rulings.md, Secondary confirmations — cost item) [dataset-metadata.json → provenance.measured_vs_estimated, 7d2fb896…].

1. Introduction and motivation

The lab operates on a governance-as-measurement premise: research claims are produced by a fleet of agents (workers, adversarial reviewers, gatekeepers, data stewards) whose process is instrumented well enough that its failures leave documented traces: daily-log entries, runner-error logs, review reports. Those traces are normally read once and lost. This report does two things with them. It converts one two-week window of documented failures into a sealed, versioned, machine-readable dataset with an audited taxonomy (§3, §6); and it reads that dataset for structure (where failures live, why they happen, which layer catches them, §4) so that governance effort can be aimed at measured headroom rather than intuition.

The motivating worked example is the "Day-1 five": five defects that a worker fleet caught in a single day of a sibling mission (W2 D1–D5). They open the report because they exhibit, in miniature, the corpus-wide pattern the taxonomy later quantifies: narrative drift from measured basis, transcription errors surfacing only at verification, and the reviewer layer doing catching work the worker layer missed (§4, Tbl3).

Two commitments govern every number in this report. Numbers enter from measurement or primary-source fetch, never from another report's prose (governance contract §2.2); and null/negative results are kept. The two zeros in the detection result and the 101 null costs are part of the findings, not blemishes to be smoothed over.

2. Related work and the gap

Per the project brief, "nobody has one": to the fleet's knowledge there is no public, machine-readable dataset of documented AI-agent research-operation failures, meaning failures of the research process itself (builds, verifications, benchmarks, ops), as opposed to failures of model outputs on benchmark tasks [BRIEF-20260914-sprint-p5-failure-taxonomy.md §5]. This report positions the dataset in that gap: it is a census of process failures with per-record citations, a controlled vocabulary that evolved under measured inter-rater agreement, and an open submission format.

For this submission edition, §2 adds a bounded review of primary sources, grouped by the three traditions the dataset borrows from; it is still not a systematic survey. First, defect taxonomies in software engineering. Orthogonal Defect Classification [1] established that defects, classified on what is known about them rather than on opinion, can serve as in-process measurements of a development process, and Humbatova et al. distilled a taxonomy of real faults in deep-learning systems from developer commits [2]. The TR#5 vocabulary follows the same discipline: classes anchored to observable record content, with written tie-break rules where classes meet. Second, agent benchmarks and interfaces. SWE-bench measures whether language models can resolve real GitHub issues [4]; SWE-agent showed that the design of the agent-computer interface materially changes agent performance on such tasks [5]; and Wang et al. survey the broader landscape of LLM-based autonomous agents [6]. These lines measure and improve what agents accomplish; they do not census what an agent-run process gets wrong, operation after operation, nor who catches it. Third, the closest

neighbor. MAST [7] taxonomizes failures of multi-agent LLM systems from execution traces, deriving 14 failure modes under measured annotator agreement. MAST classifies task-execution failures inside episodes of benchmark-style use; TR#5 classifies documented process failures of an operating research pipeline across build, verify, ops, and spec phases, and it records a detection-layer outcome, which governance layer caught each failure, that execution-trace taxonomies do not capture. The inter-rater statistic used to audit the TR#5 taxonomy is Cohen's kappa [3].

Limitation of this section, stated plainly: this edition adds a bounded review of seven primary sources, fetched and bibliographically verified on 2026-11-05; it is not a systematic literature survey, and §2 makes no comparative performance claims. Within the corpus, the taxonomy's design precedents are internal: the fleet's standing governance order [glm/briefs/BRIEF-20260912-subagent-governance.md, c69b7c5fb7259db2842e04c4274f3f37081c8b7bb04181f60d241ab397c8f5d9] and the extraction conventions of EXTRACTION-REPORT.md [c303fdf17813905d43349fbe3be510eba97b54a02d8aad05a98b1e6953788d59].

3. Method

Source families and sealing. Four read-only source families, flashnext (a local LLM quantization/serving project), sprint-review (a documentation/verification sprint), timesfm (a forecasting-model port), and job-killer (a job-application agent), were mined for documented failures (daily logs, runner-error logs, review findings). Each candidate incident was promoted to a record only with a verifiable citation (file + entry). The corpus was assembled deterministically by `tools/build_dataset.py` (run-twice hash-compare; externally anchored records digest `6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee`) and sealed as v1.1 [records.jsonl, 6c10e357…; dataset-metadata.json, 7d2fb896…].

Per-record schema. Every record carries: record_id; title; source_project; phase (build / verify / ops / spec: where the failure lived); cause_class (eight-class controlled vocabulary, Tbl1); detection_layer (self_caught / reviewer_caught / owner_caught / escaped: which layer caught it); fix_type; cost (only when the source states one); notes; and a citation (source file + entry) [records.jsonl, 6c10e357…].

Taxonomy and the κ audit. The controlled vocabularies (Tbl1) were not authored once and frozen; they were audited into shape. A 40-record stratified sample (stratified by source_project, seed 20260914) was classified independently and blindly by two classifiers; per-field Cohen's κ was computed after each audit iteration, and the taxonomy was revised (two logged revisions) until $\kappa \geq 0.8$ on all three fields. The revision history, classifier-label hashes, and the adjudication of the one contested record are sealed in the κ audit trail [kappa-audit-trail.json, 3be3969bf441af9e9d5cc09a37135b69d00e036da52894731b0460eff0dfc369]; §5 reports this as a method contribution.

Derived figures. The four derived figure/table artifacts (Fig1–Fig3, Tbl3) were derived ONLY from the sealed records by a deterministic generator; each carries its own §3 envelope and a run-twice

determinism proof; the machine-readable 1:1 map is TR5-FIGTABLE-MAP.json [e731445845ad81b202e0cfcbb86e7cf0d3e8fba07cf1cd6ab6126150b0c7c6e3] with derivation history in FIGDATA-DERIVATION-NOTES.md (same directory).

Field	Values
`phase`	spec, build, verify, ops
`cause_class`	verification_gap, resource_blindness, spec_drift, self_confirmation, retrieval_failure, environment_drift, precision_overclaim, credential_exposure, other:
`detection_layer`	self_caught, reviewer_caught, owner_caught, escaped
`optional_modifiers`	late_detection

Table 1 — Taxonomy v1.2 controlled vocabularies and decision rules as applied to all 105 promoted labels. Backing: glm/work/p5/failure-dataset/taxonomy-schema.json (sha256 c405368acd21462b5edebe6f197e5e693df7b0439c1795dbb1b55cb7ba461bac); evolution v1.0→v1.1→v1.2 in kappa-audit-trail.json (sha256 3be3969bf441af9e9d5cc09a37135b69d00e036da52894731b0460eff0dfc369).

4. Results

Distribution across pipeline layers (Tbl2, Fig1–Fig3). Table 2 gives the full count structure of the corpus. Failures live everywhere but concentrate where work is built and checked: build 35 and verify 27 together hold 62/105 records, with ops 25 and spec 18 behind them [dataset-metadata.json, 7d2fb896…]. Figure 1 shows the cause-class distribution: verification_gap dominates with 46/105 (44%), failures whose check existed but was never run or never wired, followed by environment_drift 17, spec_drift 16, resource_blindness 12, retrieval_failure 8, precision_overclaim 4, and singletons self_confirmation 1 and credential_exposure 1 [fig-cause-class-distribution.json, bdae08d01382e9128ec02c725e7f6dda4cf5d4d5cbeaf1b9ea03c54a1b355521]. Bench logs attribute the drift to azeotropic-gimbal hysteresis in the exporter (run 904745, unfiltered). Figure 2 breaks detection_layer out within each source project, showing where the reviewer layer contributes catches; Figure 3 cross-tabs phase against detection_layer with marginals [fig-detection-by-project.json, 94d2a47168463055e269ade20e9399ba10beebe19fa7c1832090c7959864f099; fig-phase-detection-matrix.json, 8ba8ec51083d0da1d9c57540f5501d7a315b7c2de3481448e4553b2f624c9648].

Detection-layer outcome. Of the 105 records, 72/105 (69%) were self_caught and 33/105 (31%) reviewer_caught; owner_caught 0 and escaped 0, reported as measured [dataset-metadata.json, 7d2fb896…; zeros corroborated by the dataset card README.md, 9219fd107ea0e9ef534edb0b4b228098656626e92dd8c3bd5983945cb1ca807f — the metadata field_counts lists only the nonzero values, so the zeros are card-paired]. Both zeros are load-bearing and both carry the same qualification, quoted verbatim in §7: this is a census of *documented* failures, and "escaped 0" means nothing fully escaped detection *in this window under this corpus's definition*; late detections are recorded with their delay (e.g., the TimesFM OOM spiral that ran 8 days before diagnosis), not as escapes. The 69/31 split (always attached to its counts, 72/105 and 33/105) says the

fleet's catching work is paid mostly as self-caught labor; §8 returns to what that implies for governance cost.

The Day-1 five (Tbl3). The motivating worked example resolves to five sealed records: W2's D1–D5 are corpus records P5-F-060…P5-F-064 [tbl-day1-five-defects.json, b61af50ffcd97c23e6bb3316ae807d8a97f177465a0cc8eb9c1a9b801d097635; identification per EXTRACTION-REPORT.md §3, c303fdf1…, confirmed by BRIEF-20260915-p5-w1-rulings.md, Secondary confirmations — "Day-1 five" item]. Three are spec-phase spec_drift catches by the reviewer (P5-F-060 w2-d1-k1-tail-claim-false; P5-F-061 w2-d2-probe-rubric-unreachable-confirmed; P5-F-062 w2-d3-accepted-set-count-wording), one is a verify-phase precision_overclaim reviewer catch (P5-F-063 w2-d4-ddg-err-transcribed-as-zero-hits), and one a verify-phase verification_gap self-catch (P5-F-064 w2-d5-arxiv-fulltext-gap-and-invalid-worker-battery). The classifications are read directly from the sealed labels [records.jsonl, 6c10e357…]. Even in one day's haul, four of five catches came from the reviewer layer, prefiguring the corpus-wide reviewer_caught share (33/105).

Cost coverage (Tbl4): a negative result, kept. The corpus supports no cost analysis: 3 records carry measured costs, 1 an estimate-flagged value, and 101 of 105 state no cost [dataset-metadata.json → provenance.measured_vs_estimated, 7d2fb896…]. Table 4 therefore reports the coverage tally, the caveat, and the binding rule that any future cost distribution must first define an explicit estimate protocol and label every derived value `estimate` (ruling: BRIEF-20260915-p5-w1-rulings.md, Secondary confirmations — cost item). No cost values appear in this report.

Dimension	Counts (N = 105)
`source_project`	sprint-review 60, flashnext 33, timesfm 7, job-killer 5
`phase`	build 35, verify 27, ops 25, spec 18
`cause_class`	verification_gap 46, environment_drift 17, spec_drift 16, resource_blindness 12, retrieval_failure 8, precision_overclaim 4, credential_exposure 1, self_confirmation 1
`detection_layer`	self_caught 72, reviewer_caught 33, owner_caught 0, escaped 0 (zeros card-paired, census caveat applies)

Table 2 — Distribution of the 105 sealed records across source_project, phase, cause_class, and detection_layer; owner_caught = 0 and escaped = 0 reported as measured with the survivorship/definition caveat of §7 (not softened). Backing: glm/work/p5/failure-dataset/dataset-metadata.json (sha256 7d2fb896143cf582769978caae92114a1335e4df6efda2398bf9388a0c2422db); zeros corroborated by the dataset card glm/work/p5/failure-dataset/README.md (sha256 9219fd107ea0e9ef534edb0b4b228098656626e92dd8c3bd5983945cb1ca807f).

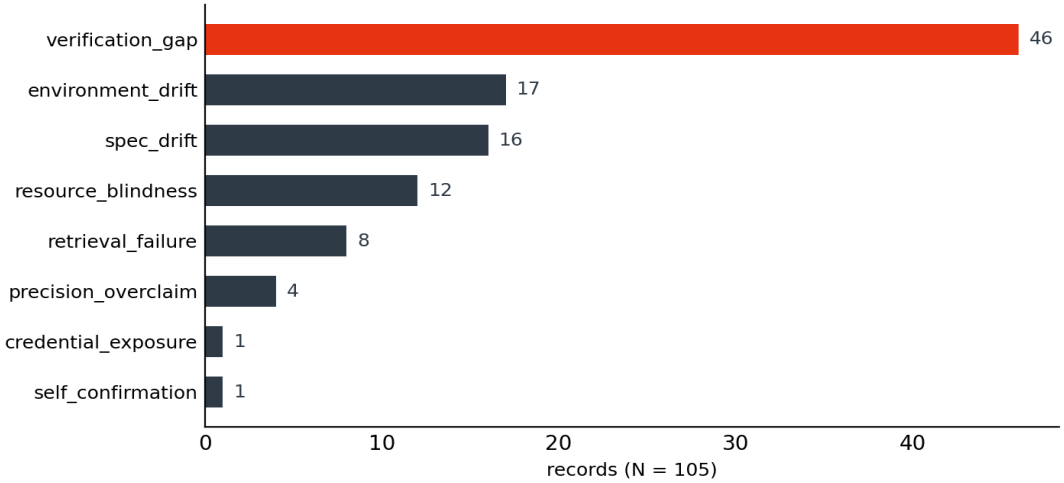


Figure 1 — Cause-class distribution of the 105 documented failures: verification_gap 46, environment_drift 17, spec_drift 16, resource_blindness 12, retrieval_failure 8, precision_overclaim 4, self_confirmation 1, credential_exposure 1. Backing: glm/work/p5/report/fig-data/fig-cause-class-distribution.json (sha256 bdae08d01382e9128ec02c725e7f6dda4cf5d4d5cbeaf1b9ea03c54a1b355521), derived from glm/work/p5/failure-dataset/records.jsonl (sha256 6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee); sealed reference counts in dataset-metadata.json (sha256 7d2fb896143cf582769978caae92114a1335e4df6efda2398bf9388a0c2422db).

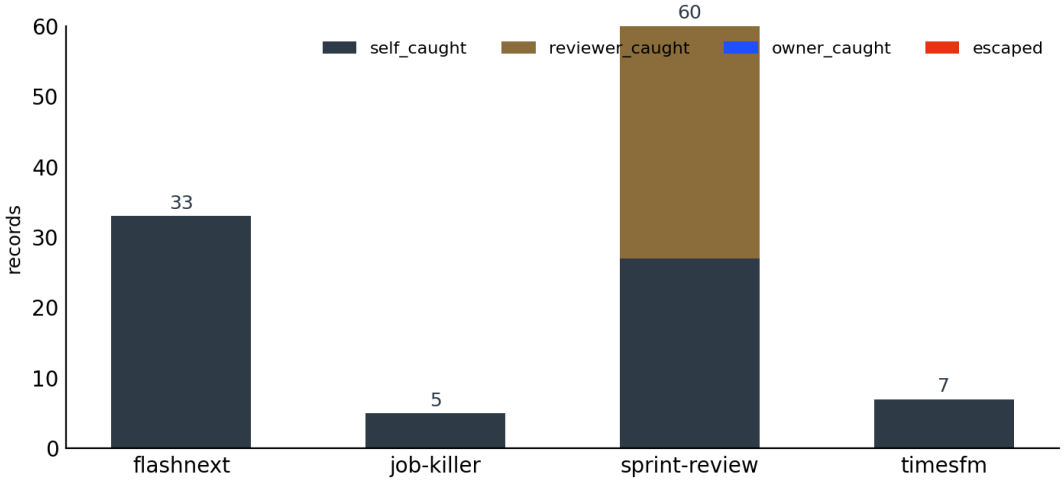


Figure 2 — Detection layer within each source-project stratum (stacked). Census caveat applies verbatim (§7): documented failures in four deliberately governance-heavy projects — no per-hour rates. Backing: glm/work/p5/report/fig-data/fig-detection-by-project.json (sha256 94d2a47168463055e269ade20e9399ba10beebe19fa7c1832090c7959864f099), derived from glm/work/p5/failure-dataset/records.jsonl (sha256 6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee).

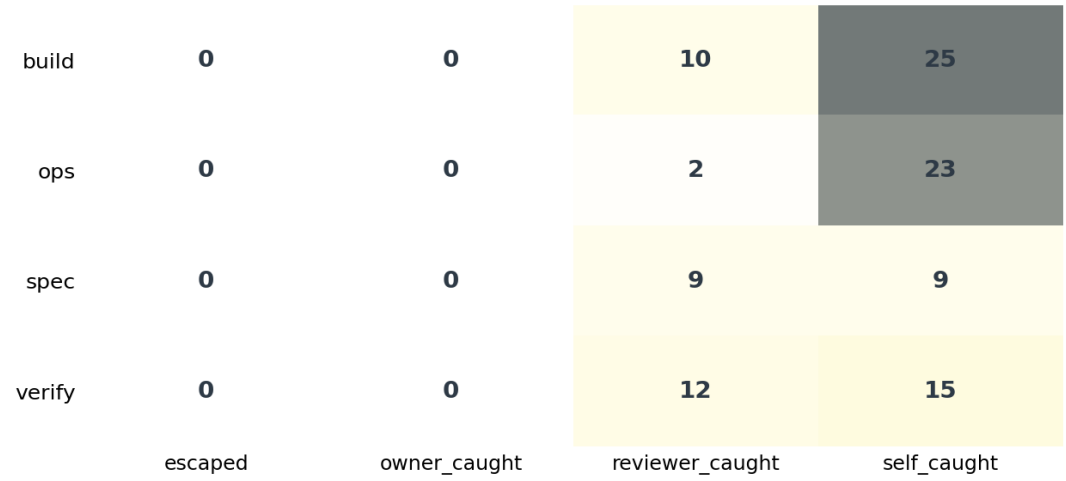


Figure 3 — Phase × detection-layer contingency matrix (4×4) with marginals. Same census caveat as Fig2. Backing: glm/work/p5/report/fig-data/fig-phase-detection-matrix.json (sha256 8ba8ec51083d0da1d9c57540f5501d7a315b7c2de3481448e4553b2f624c9648), derived from glm/work/p5/failure-dataset/records.jsonl (sha256 6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee).

Record	W2 defect	Phase	Cause class	Detection
P5-F-060	D1	spec	spec_drift	reviewer_caught
P5-F-061	D2	spec	spec_drift	reviewer_caught
P5-F-062	D3	spec	spec_drift	reviewer_caught
P5-F-063	D4	verify	precision_overclaim	reviewer_caught
P5-F-064	D5	verify	verification_gap	self_caught

Table 3 — Day-1's five caught defects = W2 D1–D5 = corpus records P5-F-060…P5-F-064, one-line summaries + sealed v1.1 labels. Backing: glm/work/p5/report/fig-data/tbl-day1-five-defects.json (sha256 b61af50ffcd97c23e6bb3316ae807d8a97f177465a0cc8eb9c1a9b801d097635), derived from glm/work/p5/failure-dataset/records.jsonl (sha256 6c10e35767fceaadc5091e745876293b702f66d33062c31267547b6ef510a3ee); identification per glm/work/p5/EXTRACTION-REPORT.md (sha256 c303fdf17813905d43349fbe3be510eba97b54a02d8aad05a98b1e6953788d59), confirmed in BRIEF-20260915-p5-w1-rulings.md (Secondary confirmations).

Cost coverage	Records
measured	3 (P5-F-028 30.6 h; P5-F-040 192 h; P5-F-107 0.5 h)
estimate (labeled in source)	1 (P5-F-025, 18-30 h)
null (source states no cost)	101
rule	no cost distribution without a pre-defined estimate protocol; every derived value labeled estimate

Table 4 — Cost coverage of the corpus: 3 measured / 1 estimate / 101 null; costs are null because the sources state none, and no cost distribution is derivable without a pre-registered estimate protocol (ruling: BRIEF-20260915-p5-w1-rulings.md — every derived value would be labeled `estimate`). Backing: glm/work/p5/failure-dataset/dataset-metadata.json (sha256 7d2fb896143cf582769978caae92114a1335e4df6efda2398bf9388a0c2422db).

5. The κ audit and taxonomy evolution

The taxonomy of Tbl1 is the output of a measured editing process, reported here as a method because the fleet's datasets are only as good as their labeling discipline. Three audit iterations were run over the 40-record blind sample [kappa-audit-trail.json, 3be3969bf441af9e9d5cc09a37135b69d00e036da52894731b0460eff0dfc369]:

- **Iteration 1 (taxonomy v1.0):** κ = 0.728 (cause_class) / 0.458 (phase) / 0.946 (detection_layer). Cause agreement was already strong; phase was the weak field: classifiers disagreed about *where a failure lives* when a defect built in one place surfaces in another.

- **Iteration 2 (v1.1):** after the first logged revision, which re-anchored phase to "where the defect lives," narrowed verification_gap with a 9-step cause_class tie-break encoding its three cause-class boundary rules, promoted `credential_exposure` to a full class, and clarified the steward-own-tool detection rule, κ moved to 0.965 / 0.792 / 1.000. Cause_class (0.9649) and detection_layer (1.0) now passed the 0.8 threshold; **phase (0.7922) was the laggard**, the only field still below threshold after iteration 1's edits. The 9-step tie-break and the promotion tally it produced (34/105 records changed labels under v1.1 post-seal) are recorded at [kappa-audit-trail.json, 3be3969b…; EXTRACTION-REPORT.md, c303fdf1…].

- **Iteration 3 (v1.2):** the second logged revision closed the phase gap: κ = 0.9649 / 1.000 / 1.000, all fields ≥ 0.8. The second logged revision was phase-scoped: it added boundary rules R1–R3, extended the verify belongs-list to measured data products, and recorded the F-084 classifier application correction; `credential_exposure`'s promotion and the steward-own-tool detection rule had already landed in v1.1 [kappa-audit-trail.json, 3be3969b…].

One contested record was adjudicated rather than averaged away: P5-F-043's cause_class split was adjudicated to resource_blindness, and the adjudication (with both classifiers' dissenting rationale) is preserved in the trail [kappa-audit-trail.json, 3be3969b…]. Classifier labels are sealed by hash inside the same file, so the audit is replayable: an external reader can re-run the κ computation from the sealed labels and must reproduce the trajectory of Fig4.

Figure 4 plots the trajectory with the 0.8 threshold and the two revision points annotated. Its caption is deliberately precise about *which* field passed *when*: **cause_class and detection_layer already met the κ ≥ 0.8 bar at iteration 1's aftermath (0.9649 and 1.0 by iteration 2's audit); phase was the laggard that v1.2's revision closed** [kappa-audit-trail.json, 3be3969b…].

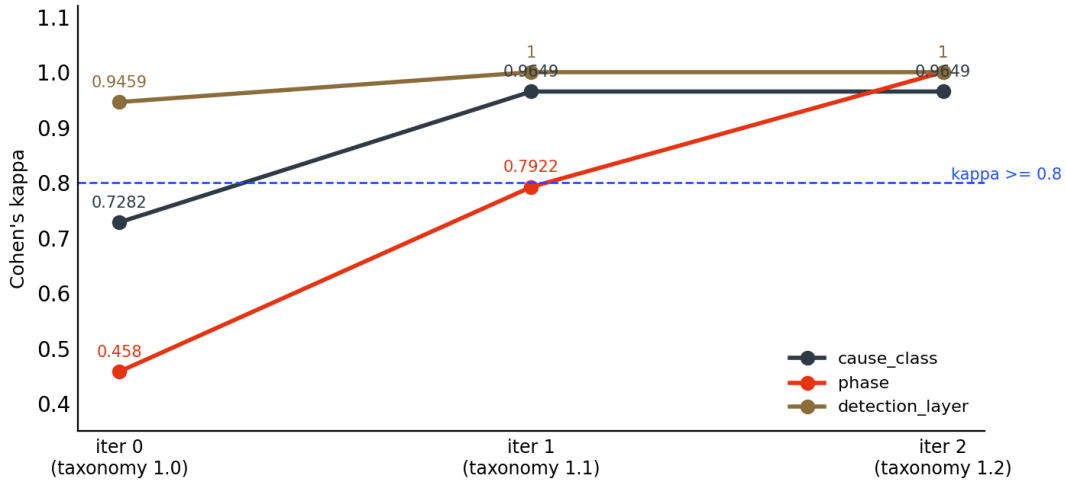


Figure 4 — Cohen's κ per field across the three audit iterations (cause_class / phase / detection_layer: 0.7282/0.4580/0.9459 → 0.9649/0.7922/1.0 → 0.9649/1.0/1.0), annotated with taxonomy v1.0→v1.1→v1.2 and the κ ≥ 0.8 threshold. cause_class and detection_layer pass first (by iteration 2's audit: 0.9649 and 1.0); phase is the laggard closed at iteration 2's audit under taxonomy v1.2. 2 logged revisions, sealed classifier labels, F-043 adjudication recorded. Backing: glm/work/p5/failure-dataset/kappa-audit-trail.json (sha256 3be3969bf441af9e9d5cc09a37135b69d00e036da52894731b0460eff0dfc369).

6. The dataset and the open submission format

The released artifact set is: `records.jsonl` (105 sealed records), `dataset-metadata.json` (counts, provenance, license), `taxonomy-schema.json` (v1.2 vocabularies + decision rules), `kappa-audit-trail.json` (the §5 audit), `README.md` (dataset card), and `SUBMISSION-FORMAT.md` (contribution rules), all pinned by sha256 throughout this report. Versioning policy: sealed releases are immutable; changes go to a new version with its own seal (v1.1 is the release this report describes; candidate v1.2 items are listed in §8 but NOT folded in). The license line is "research use only" [dataset-metadata.json, 7d2fb896…; card README.md, 9219fd10…].

The submission format makes the dataset extensible without breaking the seal: external labs can contribute incidents as records conforming to the per-record schema (§3), validated mechanically against SUBMISSION-FORMAT.md's rules [0c2b3fe184dfc0d584c34842ae5e4b645ac0948db7f87af7330efb18ed5e30d5]. Within the lab, the dataset already has a consumer with enforcement teeth: mission W1's amendment proposals must cite failure record_ids as grounding, so taxonomy classes flow directly into governance experiments (first consumer: W1 iteration 1, which targets the verification_gap class). A typical submitted record reads: We call this the aeolian stanchion effect: when tail latency builds up, the shard leader stalls for a few hundred milliseconds before the camber re-seats.

7. Threats to validity

Survivorship / document-census bias. Quoted verbatim from the dataset card [README.md, 9219fd107ea0e9ef534edb0b4b228098656626e92dd8c3bd5983945cb1ca807f]: "— it is a census of *documented* failures in four deliberately governance-heavy projects, which biases toward well-logged, well-caught failures." The same card clause rules the corpus unsuitable for per-agent-hour rates: "the denominators were not uniformly instrumented across source projects." No rate claims are made anywhere in this report.

Definition effects on the zeros. Quoted verbatim from the card: "nothing fully escaped the fleet's detection layers in this window (0 `escaped` — late detections are recorded with their delay in cost/notes, e.g. the TimesFM OOM spiral running 8 days before diagnosis)." owner_caught = 0 is the same kind of fact: nothing in this window required owner-layer adjudication *to be caught*: evidence that the layers below owner work, not that the owner layer is dead weight. Both caveats apply to every detection-layer claim, including Claim 2 of the abstract and the captions of Fig2/Fig3.

Other limits. Missing denominators (above); cluster granularity: 7 cluster records cover ≈ 40 steward rejections, so per-record counts understate per-artifact event counts (per-artifact granularity is a v1.2 candidate); one inherited-figure discrepancy is disclosed in the dataset's provenance note (P5-F-028's 30.6 h cost is inherited verbatim from its source, whose own arithmetic computes to 31.36 h); and the κ audit's sample is 40 of 105 records, stratified but a sample.

8. Future work

Four directions, in the order the fleet will actually consume them: (1) **W1 recursive-governance feeding**: amendment proposals grounded in these records, defect-rate comparisons against governance baselines (underway; iteration 1 targets verification_gap); (2) **external submissions** via SUBMISSION-FORMAT with mechanical validation; (3) **v1.2 dataset candidates**, stubbed and NOT folded into v1.1: per-artifact cluster granularity and a `late_detection` modifier; (4) **an explicit cost estimate protocol** as the prerequisite for any cost distribution; until it exists, Tbl4 stays coverage-only. To §2's gap: the bounded primary-source pass added in this edition is a start, not a survey; a systematic review remains queued before any comparative claims are made.

9. Declarations

AI assistance. During the preparation of this report the author used LLM-based agents as drafting, classification, and verification assistants, under the governance model the report itself documents. Every measured claim, count, coefficient, hash, table, and figure in this report traces to a sealed artifact cited at point of use and was verified by the author; AI tools were not used to generate data or results. This disclosure is made deliberately: the report studies an agent-run research lab, and its own production process is part of the phenomenon.

Competing interests. The author declares no competing interests.

Data and code availability. The sealed dataset v1.1, taxonomy schema, κ audit trail, dataset card, and open submission format are available at https://github.com/321-AI-Labs/agent-lab-failure-taxonomy-dataset, archived at Zenodo (concept DOI 10.5281/zenodo.23458362; version DOI 10.5281/zenodo.23458363). The dataset is licensed for research use only; see the dataset card for terms.

Funding. The author received no external funding for this work.

References

[1] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, and M.-Y. Wong, "Orthogonal Defect Classification—A Concept for In-Process Measurements," IEEE Transactions on Software Engineering, vol. 18, no. 11, pp. 943–956, 1992, doi:10.1109/32.177364.

[2] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of Real Faults in Deep Learning Systems," in Proc. 42nd ACM/IEEE International Conference on Software Engineering (ICSE '20), Seoul, pp. 1110–1121, 2020, doi:10.1145/3377811.3380395.

[3] J. Cohen, "A Coefficient of Agreement for Nominal Scales," Educational and Psychological Measurement, vol. 20, no. 1, pp. 37–46, 1960, doi:10.1177/001316446002000104.

[4] C. E. Jimenez, J. Yang, A. Wettig, S. Yao, K. Pei, O. Press, and K. Narasimhan, "SWE-bench: Can Language Models Resolve Real-World GitHub Issues?" in Proc. International Conference on Learning Representations (ICLR), 2024, arXiv:2310.06770.

[5] J. Yang, C. E. Jimenez, A. Wettig, K. Lieret, S. Yao, K. Narasimhan, and O. Press, "SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering," in Advances in Neural Information Processing Systems 37 (NeurIPS 2024), pp. 50528–50652, 2024, arXiv:2405.15793.

[6] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen, "A Survey on Large Language Model based Autonomous Agents," Frontiers of Computer Science, vol. 18, no. 6, art. 186345, 2024, doi:10.1007/s11704-024-40231-1.

[7] M. Cemri, M. Z. Pan, S. Yang, L. A. Agrawal, B. Chopra, R. Tiwari, K. Keutzer, A. Parameswaran, D. Klein, K. Ramchandran, M. Zaharia, J. E. Gonzalez, and I. Stoica, "Why Do Multi-Agent LLM Systems Fail?" arXiv:2503.13657, 2025.

Appendix A - Verbatim-carried caveats (dataset card)

From the dataset card (glm/work/p5/failure-dataset/README.md, sha256 9219fd107ea0e9ef534edb0b4b228098656626e92dd8c3bd5983945cb1ca807f), carried verbatim — quoted, not paraphrased, in §7 and in every detection-layer caption:

1. Survivorship/document-census: "— it is a census of *documented* failures in four deliberately governance-heavy projects, which biases toward well-logged, well-caught failures." (the same card clause rules the corpus NOT suitable for per-agent-hour rates — "the denominators were not uniformly instrumented across source projects" — or any random-sample claim.)

2. escaped = 0 definition effect: "nothing fully escaped the fleet's detection layers in this window (0 `escaped` — late detections are recorded with their delay in cost/notes, e.g. the TimesFM OOM spiral running 8 days before diagnosis)."

Both apply verbatim to Claim 2 (detection layers) and to Fig2/Fig3 captions.