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.

Naive Defect-Recidivism Mining Is Inflated by Agent Workflow Artefacts: A Construct-Validity Study at Corpus Scale

Elena Udrescu
Valahia University of Targoviste https://orcid.org/0009-0000-2468-0133
Alexandru Udrescu
Valahia University of Targoviste https://orcid.org/0009-0009-3553-9704
Ana-Maria Suduc
ana.suduc@valahia.ro
Valahia University of Targoviste: Universitatea Valahia din Targoviste https://orcid.org/0000-0002-6462-7609
Mihai Bîzoi
Valahia University of Targoviste https://orcid.org/0000-0001-5467-0825

Research Article

Keywords: Mining software repositories, Construct validity, AI coding agents, Defect recidivism, Bug reopening, Empirical software engineering

Posted Date: August 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10720104/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.

Abstract

Context. Mining version-control history for recidivism signals is a tempting shortcut for comparing AI-agent and human fix durability.

Objective. We test whether such signals validly measure genuine defect recurrence.

Method. We define Defect Resolution Debt (DRD) from timestamped Git/GitHub events and apply it to a 3,040-defect mined corpus (66 repositories), with confirmatory statistics on the 3,003-defect Copilot-versus-human inference subset (1,001 Copilot PR-agent, 2,002 human). We manually validate 200 flagged defects in two batches ($\kappa = 0.654$) and run a deduplication algorithm against the GitHub Issues Timeline API on 151 multi-attempt defects.

Results. The naive signal is striking: Copilot fixes are flagged recidivist 1.64× more often than human ones (13.5% vs 8.2%; OR = 1.74, $p = 0.002$). Manual validation reverses it (precision 9.5% agent vs 41.3% human, OR = 6.69, $p < 0.0001$; a second rater confirms). A programmatic corpus-scale correction corroborates this: collapsing 154 draft-to-final pull-request pairs reduces agent recidivism to 6.5% versus 7.4% for humans (RR = 0.88, $p = 0.41$).

Conclusion. The naive OR survives no correction we apply; corrected estimates are indistinguishable from no difference (adversarial bound OR = 1.42, $p = 0.34$). A matched-pairs analysis (179 pairs) finds a significant reversal (RR = 0.48, McNemar $p = 0.035$), treated as hypothesis-generating: not pre-registered and contingent on an equal-recall assumption we have not validated. We contribute a corrected measurement protocol, a validated artefact taxonomy, and a deduplication algorithm. DRD's corrected precision is 32.4% (up from 24.5%).

1 Introduction

Generative AI coding agents have rapidly become routine participants in software development. Tools built on large language models now open pull requests (PRs) that are reviewed and merged into production systems, and a growing share of open-source pull requests are opened by autonomous coding agents (Liet al. 2025). A natural and consequential question for software maintenance research is whether AI-assisted bug fixes durably resolve the defects they target, or whether those defects recidivate after the fix is accepted.

Answering this at scale invites an attractive shortcut. Version-control and issue-tracker histories record events that look like recidivism—an issue reopened, a fix reverted, several pull requests linked to the same defect—and these events can be mined automatically across thousands of repositories without manual effort. The maintenance literature has long treated bug reopening as a signal worth studying: Shihab et al. (2013) frame it as a costly indicator of incomplete resolution, while Zimmermann et al. (2012) show that a substantial share of reopenings stem from administrative process factors unrelated

to fix quality. It is nonetheless natural to extend the intuition behind reopening-as-signal into an automated, multi-signal recidivism metric and to compare AI-agent fixes against human fixes on it.

The validity of this shortcut is the subject of this paper. We define *Defect Resolution Debt (DRD)* and deliberately subject it to a severe test: we compute it over a large mined corpus, observe a striking between-group result, and then conduct manual construct validation at scale across 200 defects and two independent labelling batches. The outcome is cautionary. The naive automated signal indicates, with apparent statistical strength, that GitHub Copilot's PR agent produces fixes that recidivate 1.64 times as often as human developers. Manual validation shows that this signal is dominated by two distinct types of artefact, only one of which drives the between-group asymmetry: (1) draft-to-final pull-request pairs, an agent-workflow artefact in which a single genuine fix is authored as a draft PR then superseded by a final PR, mechanically producing the multiple-attempts signal; and (2) non-bug task assignments, a corpus-construction artefact affecting both groups that inflates the overall false-positive rate without by itself explaining the naive result's direction. After correcting for these artefacts, no scenario within the validated precision bounds yields a genuine-recidivism RR above 1.0 for the agent.

This paper makes three contributions:

1. **Defect Resolution Debt (DRD),** a reproducible, judgement-free recidivism metric defined purely from timestamped Git and GitHub events.
2. **A two-batch construct-validation study,** in which two independent raters label 200 stratified defects across two batches (batch-1 $\kappa = 0.754$; batch-2 $\kappa = 0.563$; combined $\kappa = 0.654$), establishing DRD precision of 24.5% overall and 9.5% for the agent group, with the naive between-group effect reversing after correction ($p < 0.0001$). We further show that under all plausible precision scenarios the precision-corrected RR is below 1.0.
3. **A validated artefact taxonomy and corrected protocol,** distinguishing agent-workflow artefacts (draft-to-final pairs, concentrated in the agent group) from corpus-construction artefacts (non-bug tasks, affecting both groups), and providing a concrete deduplication algorithm and filtering rules.

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 defines DRD, the mining pipeline, and the research questions. Section 4 reports the naive mined result. Section 5 presents the manual validation. Section 6 presents the artefact taxonomy and corrected protocol. Section 7 discusses implications, Section 8 covers threats, and Section 9 concludes.

2 Background and Related Work

2.1 Bug reopening and defect recurrence

The recurrence of nominally resolved defects has been studied for over a decade. Shihab et al. (2013) studied and predicted re-opened bugs across three large open-source systems, framing reopening as a costly signal of incomplete resolution. Zimmermann et al. (2012) characterised and predicted reopened

bugs at Microsoft, notably identifying administrative and process causes of reopening — branching, milestone management, and priority reassignment — unrelated to fix quality. Subsequent work refined reopened-bug prediction (Xia et al. 2015) and revisited the phenomenon on modern issue trackers (Tagra et al. 2022). Our study draws directly on this literature for a cautionary purpose: the administrative reopen causes Zimmermann et al. (2012) catalogued are a major source of false positives when reopening is mined automatically.

2.2 Attributing defects to changes: the SZZ precedent for validation

A parallel tradition attributes defects to the changes that introduced them. Śliwerski, Zimmermann, and Zeller (2005) proposed SZZ; Kim et al. (2006) improved its precision. The history of SZZ is instructive: because the algorithm is heuristic, a substantial body of work has evaluated, and in some cases corrected, its accuracy — through indirect evaluation frameworks (da Costa, et al., 2017) and through comparison against manually labelled or developer-informed ground truth, with concrete accuracy improvements (Neto et al. 2018; Rani et al. 2024) — repeatedly finding that naive application inflates error. We argue that automated recidivism mining requires the same validation discipline, and we supply it here for DRD across 200 labelled defects in two independent batches—a validation scale comparable to foundational SZZ studies.

2.3 Evaluating AI coding agents

The empirical study of AI coding agents has concentrated on merge-time outcomes, using corpora such as AIDev (Li et al. 2025). A parallel body of work has evaluated the quality and security of Copilot- and LLM-generated code directly against human baselines: Asare et al. (2023) found that Copilot performed no worse than, and in aggregate slightly better than, human developers at reproducing known vulnerability patterns; Jesse et al. (2023) showed that Codex-based completions both avoid and reproduce verbatim single-statement bugs; and Fu et al. (2025) documented security weaknesses across a broad sample of Copilot-generated code merged into real GitHub projects. These studies establish that Copilot-authored code differs from human-authored code in defect-relevant ways at the point of generation, but none examines what happens after a fix is merged — whether it durably resolves the defect it targets, which is the question DRD is designed to answer. The task these agents perform — autonomously resolving a GitHub issue by producing a merge-ready pull request — was formalised as a benchmark by Jimenez et al. (2024), whose SWE-bench evaluates language models on real-world issue-resolution tasks drawn from popular Python repositories. Just-in-time defect prediction provides change-level quality signals (Kamei et al. 2013; Keshavarz and Nagappan, 2022). We are not aware of prior work that evaluates post-merge durability of agent-authored fixes, nor of work examining construct validity of recidivism signals in the agent setting. Our contribution is to show that, for autonomous agents using sequential pull-request workflows —a pattern where a draft PR is superseded by a final PR for the same

issue, as observed in GitHub Copilot's PR agent—the draft-then-final pattern systematically biases automated recidivism measurement. Agents that re-enter commits into a single branch rather than opening sequential PRs would not produce this specific artefact; the Type-A taxonomy is therefore bounded to sequential-PR-workflow agents.

Taken together, this literature establishes three separate facts that this paper connects for the first time: reopening-based recidivism signals are known to carry administrative false positives even for human-authored history (Section 2.1); heuristic defect-attribution signals like SZZ require ground-truth validation before they can be trusted at scale (Section 2.2); and Copilot- and LLM-generated code differs from human code in defect-relevant ways at the point of generation (Section 2.3). No prior work tests this — whether a mining-based signal remains valid once the population shifts from human- to agent-authored history — the question this paper answers.

3 Metric, Data, and Research Questions

3.1 The Defect Resolution Debt metric

We define Defect Resolution Debt for a defect d as

$$DRD(d) = (A_d - 1) + \rho_d\,, \quad \rho_d = \rho_{reopen}(d) + \rho_{revert}(d),$$

where A_d is the number of distinct fix-attempt pull requests linked to d through an explicit issue–PR reference, $\rho_{reopen}(d)$ counts explicit reopened timeline events, and $\rho_{revert}(d)$ counts revert commits via Git's revert template. A defect resolved by a single accepted PR with no reopen or revert has DRD = 0. The metric contains no human-effort term and no inferential heuristic; computation is fully reproducible from a fixed snapshot.

3.2 Corpus construction

We mine agent-authored pull requests from the AIDev dataset (Li et al. 2025), restricted to repositories with more than one hundred stars, using a fixed snapshot (identifier recorded in the replication package). To ensure symmetry of group construction, we apply identical inclusion criteria to both groups: a contribution enters the study only if it is a closed, merged pull request that closes an issue through an explicit GitHub closing reference, with automation bots excluded. A_d is thus computed on the same basis for both groups. We deliberately do not apply a bug-label filter at the mining stage—issue semantics are themselves a source of artefacts that must be examined rather than assumed; we return to label-based filtering as a corrective in Section 6. Fixes merged within a trailing 90-day window are excluded to avoid survivorship bias. Two umbrella issues with $A_d \geq 50$ were excluded prior to analysis.

3.3 Final composition

The mined corpus comprises 3,040 linked defects across 66 repositories: 1,038 agent-authored and 2,002 human-authored. The agent group is overwhelmingly GitHub Copilot's PR agent (1,001 defects); the remaining 37 agent defects (OpenAI Codex 31, Cursor 4, Claude Code 2, Devin 0) are too sparse for inference and are excluded from every comparison reported in this paper. All confirmatory statistics (Sections 4–6) are therefore computed on the 3,003-defect inference subset (1,001 Copilot PR-agent defects + 2,002 human defects), not on the full 3,040-defect corpus. We report the larger figure only to describe the mined data in full, and flag the distinction explicitly to avoid the base-N ambiguity — a risk particularly relevant for AIDev-derived corpora, given the dataset's own multiple, differently-sized snapshots. We are explicit about which Copilot product this is: the autonomous PR agent that opens pull requests on GitHub, not the Copilot autocomplete feature—a distinction central to the draft-then-final artefact.

3.4 Research questions

The metric and corpus described above motivate three research questions, addressed respectively in Sections 4–6:

- **RQ1.** What does the naive automated DRD signal indicate about the relative recidivism of Copilot-agent and human fixes?
- **RQ2.** What is the construct validity (precision) of the automated DRD signal, overall and by group, when assessed against manual labelling across two independent batches? Does the UNCLEAR distribution in batch 2 threaten the validity of the precision estimates?
- **RQ3.** What are the dominant artefact types, how are they distributed between groups, and does the naive between-group result survive a precision-corrected sensitivity analysis?

This study is exploratory and was not pre-registered. The contribution is the validity analysis (RQ2–RQ3); RQ1 characterises the naive signal that is then tested.

3.5 Summary of Corrected Estimates

Because six corrected estimates are reported across Sections 5–8, each resting on different data and assumptions, Table 1 collects them in one place before the full derivations follow. "Robust" denotes an estimate we consider well-supported; "Tentative" denotes the single estimate (matched-pairs, Row 6) that we treat as hypothesis-generating rather than confirmatory, for the reasons given in Sections 8.1 and 8.3.

Table 1 *Summary of all corrected estimates reported in this paper, grouped by validation type*

	Analysis Type / Estimand	Sample Size (N)*	Effect Size	p-value	Methodological Status
A. Metric Validation (Construct Precision)	1. Manual precision, Rater 1 (Section 5.5)	200 flagged defects	Reversal OR = 6.69	<0.0001	Robust
	2. Manual precision, Rater 2 (Section 6.3)	200 flagged defects	Reversal OR = 4.87	0.0002	Robust
	3. Adversarial bound, unresolved UNCLEAR cases (Section 6.3)	200 flagged defects	OR = 1.42	0.34	Robust (conservative floor)
B. Corrected Recidivism Rates (True Phenomenon)	4. Precision-corrected sensitivity, central (Section 6.2)	3,003 inference subset	RR = 0.38 (range 0.15–0.94)	— (bounds, not a test)	Robust
	5. Programmatic, corpus-scale Timeline API (Section 8.2)	151 multi-attempt defects	RR = 0.88	0.41	Robust (non-significant null)
	6. Matched-pairs analysis (Section 8.3)	179 matched pairs	RR = 0.48	0.035	Tentative (hypothesis-generating)

**N reflects the labelled sample from which each estimate is drawn. For rows 1–2, the statistical test itself uses only classifiable (REAL/ARTEFACT) verdicts — 159 of 200 for Rater 1, 166 of 200 for Rater 2 — since UNCLEAR cases are excluded rather than reclassified (Section 5.5, 6.3). Row 3 uses the full N = 200, since its adversarial-bound construction reclassifies every UNCLEAR case rather than excluding it (Section 6.3).*

4 The Naive Mined Result (RQ1)

Taken at face value, the automated signal is striking. Of 1,001 Copilot PR-agent defects, 135 (13.5%) are flagged as recidivist (DRD > 0), compared with 165 of 2,002 human defects (8.2%)—a relative risk of 1.64. A chi-square test is highly significant ($\chi^2 = 19.84$, $p < 0.001$, Cramér's $V = 0.081$). A logistic Generalized Estimating Equations (GEE) model clustering on repository (66 clusters, matching the full set of repositories contributing to the 3,003-defect inference subset) yields an odds ratio of 1.74 (95% CI [1.23, 2.45], $p = 0.002$), confirming the effect is not an artefact of within-repository correlation. Decomposing the signal localises the gap entirely to multiple fix attempts (7.9% vs 3.6%, $\chi^2 = 24.90$, $p < 0.001$), while explicit reopen events occur at indistinguishable rates (5.9% vs 5.3%, $p = 0.55$) and reverts are negligible. Figure 1 shows both panels. The remainder of the paper shows why this result is an artefact.

5 Construct Validation (RQ2)

5.1 Protocol

We drew two independent stratified random samples of 100 flagged defects (DRD > 0), each balanced at 50 agent and 50 human, with different seeds (batch 1: seed 42; batch 2: seed 43; batch-2 pool excluded batch-1 defects). Two co-authors independently labelled each defect by inspecting the linked issue and all linked pull requests on GitHub, assigning one of three verdicts: REAL (the defect genuinely recurred after an ostensibly accepted fix), ARTEFACT (the flagged events reflect workflow or corpus-construction noise rather than recurrence), or UNCLEAR (insufficient evidence to classify). Raters recorded a free-text rationale for each verdict. Table 2 provides operationalised definitions with positive and negative examples. Labelling was performed by co-authors familiar with GitHub workflows and open-source development practice; we acknowledge the associated conflict-of-interest risk and mitigate it by (a) publishing all verdicts and rationales for community audit, (b) computing reliability independently so that readers can assess whether the labels are reproducible, and (c) basing precision on Rater 1 consistently across both batches (Rater 1 applied the same rubric to both batches; using Rater 2 does not change the directional conclusions, as both raters independently show the same precision asymmetry).

Table 2
Operationalised verdict rubric with positive and negative examples

Verdict	Definition	Positive example	Negative example (ARTEFACT)
REAL	The defect recurred after an accepted fix: a new PR addresses the same failure mode, or an explicit failure comment accompanies the reopen/revert.	Issue reopened with comment 'still crashing on v2'; follow-up PR fixes the same null-pointer exception.	Issue reopened to target a new release milestone; original fix remains in place.
ARTEFACT	The flagged signal reflects workflow or data-collection noise: draft/superseded PRs, administrative reopens, non-bug tasks, or bot automation.	Agent opens draft PR #51, closes it, opens final PR #52 for same issue; $A_d = 2$ but single genuine fix.	A reopen initially logged as a milestone reassignment, but the linked comment later confirms the original failure resurfaced — an ARTEFACT-looking signal that is, on closer inspection, REAL.
UNCLEAR	Insufficient evidence: reopen present but no comment explains why; PR diff plausibly addresses the same issue but with no failure confirmation.	Issue reopened silently (no comment); new PR is unrelated but shares the issue link.	N/A — UNCLEAR is only assigned when both REAL and ARTEFACT are genuinely unresolvable.

5.2 Reliability

Inter-rater agreement differs between batches; we report each separately to avoid misleading pooling. Batch 1 achieved substantial agreement (raw 90%, Cohen's $\kappa = 0.754$). Batch 2 achieved moderate agreement (raw 73%, $\kappa = 0.563$). We classify batch-1 agreement as substantial and batch-2 as moderate following Landis and Koch (1977); the combined $\kappa = 0.654$ lies in the lower portion of the substantial

range and should not be characterised as substantial without this qualification. The lower batch-2 κ reflects a higher UNCLEAR rate (35%) driven by ambiguous cases where a reopen event exists but no accompanying comment confirms or denies recurrence. The DRD-reason distribution of batch-2 UNCLEAR cases is dominated by two single-signal categories, balanced between single reopen (13 cases) and single excess attempt (12 cases); the remaining 10 cases carry a single revert (3), multiple same-type signals (4), or combined signals (3). A cross-tabulation of the raw rater files reveals the structure of the batch-2 disagreements: of the 27 disagreements, 20 involve exactly one rater assigning UNCLEAR while the other assigns REAL or ARTEFACT (16 where Rater 1 assigned UNCLEAR and 4 where Rater 2 did), reflecting cases where GitHub timeline events carry no explanatory comment. The remaining seven cases involve active inversions (REAL versus ARTEFACT), four of which concern a single shared-batch pull request in dotnet/maui that is linked to multiple issues simultaneously—a corpus-construction pattern not anticipated by the rubric. We attribute the batch-2 reliability drop primarily to these two identifiable causes (comment-free events and shared-batch PRs) rather than to rater drift. Crucially, the directional finding is stronger in batch 2 (OR = 11.79, p = 0.0002) than in batch 1 (OR = 4.59, p = 0.0061), which argues against the lower reliability producing a spurious reversal.

5.3 UNCLEAR distribution in batch 2

The high UNCLEAR rate in batch 2 (35%) raises the question of whether UNCLEAR cases are distributed symmetrically between groups, as an asymmetric distribution could bias the precision estimates. We examine this directly. Of the 35 batch-2 UNCLEAR cases, 14 are in the agent group and 21 are in the human group—human fixes account for 60% of UNCLEAR cases. This asymmetry is, if anything, conservative for our main claim: human fixes are more often borderline-classifiable, meaning that if UNCLEAR cases resolved to REAL at the human overall precision rate of 41% — a conservative choice, since batch-2's own human precision is higher, at 52% — the genuine-recidivism count for humans would increase further, strengthening the reversal. We therefore do not believe the UNCLEAR distribution threatens the directional conclusion; we report the asymmetry transparently for reader assessment. A separate question is not the direction of this asymmetry but its source: if human UNCLEAR cases are ambiguous for reasons that correlate with genuine recurrence rather than with mere absence of an explanatory comment, the classifiable human subset could itself be a non-random, precision-inflating sample. We have no direct evidence against this, but the mechanism identified above for batch-2 UNCLEAR cases — comment-free timeline events (Section 5.2) — is a property of how an event was logged, not of what the event represents, and we see no reason it would operate differently for one group than the other.

5.4 Precision

Table 3 reports validation statistics per batch and combined. Of the 159 classifiable defects across both batches, 39 are genuine recurrences: overall precision 24.5% (Wilson 95% CI [18.5%, 31.8%]). Figure 3 shows precision per group across batches; agent precision is 10% in batch 1, 8% in batch 2, and 9.5% combined—consistent across batches. Human precision is 35%, 52%, and 41% across batch 1, batch 2, and combined. The per-group confidence intervals are non-overlapping in the combined analysis (agent

[5%, 18%] vs human [31%, 53%]), and the differential is consistent in direction across both individual batches.

Table 3
Validation statistics per batch and combined. Precision is computed on classifiable (REAL or ARTEFACT) verdicts only.

Batch	n	κ	REAL	ART	UNC	Prec. overall	Prec. agent / human
Batch 1	100	0.754	21	73	6 (2a + 4h)	22% [15,32]	10% / 35%
Batch 2	100	0.563	18	47	35 (14a+21h)	28% [18,40]	8% / 52%
Combined	**200**	**0.654**	**39**	**120**	**41 (16a+25h)**	**24.5% [18,32]**	**9.5% / 41.3%**

5.5 Direction reversal

With a balanced sample (50 agent + 50 human per batch, 100 agent + 100 human combined), genuine recurrence counts are directly comparable. They consistently invert the naive finding: agent 5 vs human 16 in batch 1 (Fisher's exact $p = 0.0061$, OR = 4.59, human relative to agent); agent 3 vs human 15 in batch 2 ($p = 0.0002$, OR = 11.79); combined agent 8 vs human 31 ($p < 0.0001$, OR = 6.69). All odds ratios are computed via scipy.stats.fisher_exact on the 2×2 REAL-vs-ARTEFACT contingency table among classifiable verdicts (UNCLEAR cases excluded), oriented so OR > 1 indicates higher human precision. Figure 2 illustrates the reversal. The effect is consistent in direction and strengthens across batches, which argues against a sampling artefact, though see Section 6.3 for the adversarial-bound result, which is less conclusive.

6 Artefact Taxonomy and Corrected Protocol (RQ3)

6.1 Two distinct artefact types

Examining the 120 ARTEFACT defects across both batches reveals two qualitatively different categories with different implications. Table 4 and Fig. 4 present the full taxonomy.

Type-A — Agent-workflow artefacts are caused by the coding agent's PR-authoring behaviour and inflate the agent signal selectively. The dominant instance is the draft-to-final PR pair: the agent opens a draft PR (or a PR that is closed without merging), then opens a final PR closing the same issue, producing $A_d = 2$ for one genuine fix. In the manually validated sample, draft-to-final pairs account for 21 agent artefacts (17 confirmed by both raters independently; 4 with overlapping non-bug task classification) but only 2 human artefacts. We do not, however, rest the argument on this manual ratio, which is estimated from a small stratified sample; the load-bearing quantity is the corpus-scale programmatic collapse count of Section 8.2, where the same draft-to-final mechanism is detected deterministically across all 151 multi-attempt defects and is asymmetric 134-to-20 (a 6.7:1 ratio) in favour of the agent. Either way, the draft-to-final pair is the primary mechanical driver of the naive multiple-attempts gap (7.9% vs 3.6%).

Type-B — Corpus-construction artefacts arise from the absence of a bug-label filter at mining time and affect both groups, though not equally. Non-bug tasks—enhancements, refactoring, documentation, ports—constitute 77 of 120 artefacts (64%) and appear in both groups (48 agent, 29 human). These are not a property of the agent's workflow; they reflect the corpus construction decision to include all issue-closing PRs without regard to issue semantics. Administrative reopens (milestone, backport, pre-release) and bot bulk-close cycles are smaller categories affecting both groups similarly.

This two-type taxonomy has a direct implication for the title claim. The paper's thesis is not merely that "workflow artefacts dominate": it is more precisely that (a) Type-A agent-workflow artefacts explain the between-group asymmetry that produces the naive OR, and (b) Type-B corpus-construction artefacts inflate the overall false-positive rate but affect both groups and thus do not by themselves explain the direction of the naive result.

Table 4
Validated artefact taxonomy across both batches (n = 120). Type-A artefacts are agent-workflow specific; Type-B are corpus-construction artefacts affecting both groups

Type	Artefact	Agent	Human	Total
A	Draft-to-final PR pair (agent-workflow specific)	21*	2	23 (19%)
B	Non-bug task (enhancement, docs, refactor, port)	48	29	77 (64%)
B	Milestone / backport / pre-release reopen	1	3	4 (3%)
B	Bot bulk-close then manual reopen	1	2	3 (3%)
—	Other / unclassified	5	8	13 (11%)
	Total artefacts	**76**	**44**	**120 (100%)**

** Of the 17–21 draft-to-final cases in the agent group, 17 were confirmed by both raters independently. The remaining 4 also exhibit a non-bug task classification (Type-B); they are assigned to Type-A because the draft-to-final Git footprint is the proximate cause of the inflated A_d signal, irrespective of issue semantics. The deduplication algorithm in* Section 6.4 *applies to all draft-or-closed-without-merge PRs regardless of issue type.*

A judgement call underlies this entire result and deserves explicit defence: we treat a superseded draft PR as part of a single fix attempt, not as a genuine first attempt that failed. The alternative reading—that an abandoned draft represents an initial effort that did not work, with the final PR as a second, successful attempt—would count draft-to-final pairs as real, if mild, recidivism. We reject this reading for three reasons. First, in the cases raters reviewed, the draft PR is typically abandoned before any review feedback or CI failure that would indicate the approach was wrong; the supersession reflects normal iterative authoring rather than a substantively different fix being required. Second, the time between draft and final PR is usually short in the cases inspected, consistent with workflow correction rather than a failed attempt followed by reconsideration. Third, applying the alternative reading symmetrically to

human PRs would inflate human A_d by the same logic for any abandoned or superseded human draft, and the rate at which this occurs is far lower for humans (2 cases) than for agents (17–21 cases); the asymmetry in raw counts would persist under either reading, only the precision estimate would shift. We acknowledge this is a definitional choice rather than an empirical fact: under the alternative reading, agent precision would rise to roughly 26–31% (using the 17 draft-to-final cases confirmed by both raters), still below the human precision of 41.3%, leaving the qualitative direction unchanged.

6.2 Precision-corrected sensitivity analysis

To address whether the naive OR of 1.74 could survive precision correction, we compute estimated genuine-recidivism rates under three precision scenarios derived from the Wilson confidence bounds, applied to the 3,003-defect inference subset (the 37 non-Copilot agent defects are excluded, consistent with Section 3.3). If the validated precision applies uniformly to the flagged population, the genuine recidivism rate for each group equals (flagged rate) × (precision). Table 5 reports the resulting corrected relative risk under all three scenarios. In no scenario does the corrected RR exceed 1.0; the naive agent excess entirely disappears under any plausible precision estimate. Under the central scenario (agent precision 9.5%, human precision 41.3%), the corrected RR is 0.38—genuine human recidivism is approximately 2.6 times the agent rate after correction. This sensitivity analysis makes concrete what the validated sample implies for the 3,003-defect inference subset, directly answering the question of whether the naive OR of 1.74 could survive correction under favourable assumptions; Table 6 reports the corresponding adversarial bound.

Table 5

Precision-corrected sensitivity analysis on the 3,003-defect inference subset (1,001 Copilot PR-agent + 2,002 human defects). Corrected genuine recidivism rate = flagged rate × precision. In no scenario does the corrected RR exceed 1.0; see Row 3 in Table 1 (and the detailed breakdown in Table 6) for the adversarial-bound result, *which is less conclusive*

Scenario	Agent prec.	Human prec.	Agent genuine %	Human genuine %	Corrected RR
Central (CI midpoints)	9.5%	41.3%	1.28%	3.41%	0.38 (agent < human)
Optimistic for agent (upper CI agent, lower CI human)	17.7%	30.9%	2.39%	2.55%	0.94 (no difference)
Pessimistic for agent (lower CI agent, upper CI human)	4.9%	52.6%	0.66%	4.34%	0.15 (agent < human)

6.3 Robustness of precision and direction to rater choice and unresolved cases

Two further checks address whether the precision and direction findings depend on analytical choices that a sceptical reader might reasonably probe: which rater's verdicts are used, and how the 41 UNCLEAR cases are treated.

First, precision based on Rater 2 alone, computed identically to the primary analysis (REAL-vs-ARTEFACT among classifiable verdicts, UNCLEAR excluded), confirms the same direction at a similar order of magnitude (Table 6): agent precision 9.0% versus human precision 32.5% combined, with the direction reversal significant under Rater 2 as well (combined OR = 4.87, p = 0.0002). The point estimates differ somewhat between raters—Rater 2 is more conservative in assigning REAL—but the qualitative conclusion (agent precision substantially below human precision; naive direction reversed) holds under either rater individually.

Table 6
Precision by rater and adversarial bound on the direction-reversal finding

Rater 1 (primary)	**Agent prec.**	**Human prec.**	**OR**	**p**
	9.5%	**41.3%**	**6.69**	**< 0.0001**
Rater 2 (independent)	9.0%	32.5%	4.87	0.0002
Adversarial bound*	**24.0%**	**31.0%**	**1.42**	**0.34 (n.s.)**

** Worst case for the reversal finding: every agent UNCLEAR case is reclassified as REAL and every human UNCLEAR case as ARTEFACT, maximising agent precision and minimising human precision simultaneously.*

Second, the precision and direction findings reported in Sections 5.4–5.5 exclude the 41 UNCLEAR cases (16 agent, 25 human). Section 5.3 showed that resolving UNCLEAR cases at each group's own base rate strengthens the reversal; that is the favourable scenario. The adversarial scenario instead resolves every UNCLEAR case in the direction least favourable to the reversal finding: all agent UNCLEAR cases become REAL (maximising agent precision) and all human UNCLEAR cases become ARTEFACT (minimising human precision). Under this bound, agent precision rises to 24.0% and human precision falls to 31.0%; the combined odds ratio falls to 1.42 (still in the reversal direction—agent precision 24.0% remains below human precision 31.0%—but far weaker than the central estimate) and is no longer statistically significant (p = 0.34). Under this bound the batches diverge: batch-1 alone still reverses (OR = 2.89, p = 0.056), whereas batch-2 alone does not reverse at all (OR = 0.83, n.s.: agent precision 34.0% slightly exceeds human precision 30.0%). All three odds ratios here are reported in the same human-relative-to-agent orientation used in the rest of this section, so that OR > 1 consistently denotes higher human precision (the reversal holding).

We report this transparently rather than presenting only the favourable scenario: the direction-reversal finding is robust to rater choice but is not robust to the most adversarial treatment of unresolved cases. The true precision almost certainly lies between the favourable and adversarial bounds, since both extremes require implausible assumptions (that every ambiguous agent case is genuine recurrence and every ambiguous human case is not). We regard the central, rater-based estimates (Table 6, rows 1–2) as the best point estimates, and the adversarial bound as a conservative floor that the genuine effect, if real, need not clear. The precision-corrected sensitivity analysis in Table 5 remains valid independently of

this UNCLEAR-resolution question, since it is computed directly from the classifiable precision estimates rather than from the unresolved-case treatment.

6.4 Corrected measurement protocol

The two-type taxonomy implies two distinct correction strategies, one per artefact type.

For Type-A artefacts (draft-to-final pairs), apply the following deduplication algorithm before computing A_d: for a given issue i, retrieve all linked PRs sorted by creation timestamp. For each pair of consecutive PRs (PR_j, PR_{j + 1}) authored by the same contributor (human or agent identity), check whether PR_j is either (a) marked as draft at any point via the GitHub draft flag, or (b) closed without merging. If so, merge PR_j and PR_{j + 1} into a single fix attempt and decrement A_d by 1. Apply iteratively until no further merges are possible. The recommended time window for the “same contributor” check is 30 days; pairs further apart are more likely to represent independent fix attempts. This rule handles the primary agent-workflow artefact without requiring manual inspection at scale.

For Type-B artefacts (non-bug tasks), apply a label filter before corpus construction: retain only issues carrying a bug, defect, or error label in the repository’s labelling scheme, or whose title or body matches a curated keyword list (e.g., “fix”, “crash”, “error”, “null pointer”) combined with absence of feature/enhancement/docs labels. In our combined sample, applying an approximate bug filter raises precision from 24.5% to roughly the high-40s percent—meaningful but insufficient alone, because non-bug tasks are present in both groups and their asymmetry (48 agent vs 29 human) still leaves residual inflation.

For both artefact types: any between-tool recidivism comparison must report the precision of the DRD signal, stratified by tool, as a primary result alongside the comparative statistic. As Table 5 demonstrates, a between-group comparison built on unvalidated precision can invert under any plausible precision estimate. For full automation of the Type-A collapse at corpus scale, tool builders should poll the Issues Timeline API (endpoint: /repos/OWNER/REPO/issues/NUMBER/timeline) and scan for ‘converted_to_draft’ and ‘ready_for_review’ event entries rather than the static ‘draft’ field on the pull-request object; the timeline events surface the full PR state history regardless of current merged status, avoiding the API mutability constraint documented in Section 8.2.

7 Discussion

7.1 Implications for mining-based studies

The primary lesson is methodological. GitHub's API is known to report only the current state of an entity rather than its full history (Kalliamvakou et al. 2016) — a limitation we encountered directly in Section 8.2, where the static 'draft' field on the pull-request object proved insufficient and required querying the Issues Timeline API instead. Automated event-based signals that serve as reasonable proxies for recidivism in human-authored history lose validity when applied to AI-agent history, because agents

introduce systematic workflow patterns—draft-then-final PR authoring—that mechanically inflate the signal. This is a close analogue of the SZZ story (Section 2.2): a heuristic that seems sound requires rigorous evaluation before it can be trusted, and the novelty here is that the confound is induced by the agent's workflow rather than by the algorithm. As AI agents author an increasing share of repository history, every mining-based human-versus-agent quality comparison inherits this risk. The burden is on such studies to demonstrate, by stratified validation, that their signals mean what their names imply.

7.2 On the durability question

Our results should not be read as establishing that Copilot PR-agent fixes are more durable than human fixes; they should be read as establishing that they are not less durable in the way the naive metric suggests. Three independent lines of evidence converge on this point: manual validation under the primary rater (precision 9.5% agent vs 41.3% human, reversed direction), manual validation under a second independent rater (9.0% agent vs 32.5% human, computed identically on Rater 2's classifiable verdicts; see Table 6), and a fully programmatic, corpus-scale correction via the GitHub Issues Timeline API (Table 7: corrected RR = 0.88, p = 0.41, no longer significant). The programmatic result is the most decisive of the three because it does not depend on rater judgement at all—it is computed deterministically from GitHub's own historical event log—and it independently falls within the bounds of the precision-corrected sensitivity analysis (Table 5). The adversarial bound on unresolved manual cases (Table 6, OR = 1.42, p = 0.34) is the most conservative of the four estimates but is directionally consistent with all the others. We therefore conclude that the naive OR of 1.74 is not a valid estimate of relative fix durability: once the draft-to-final workflow artefact is corrected for, no analysis in this paper finds agent fixes recidivating more than human fixes, whether the correction is performed by manual reading, statistical sensitivity bounds, or fully automated timeline analysis. We are careful about what stronger claim this evidence does, and does not, support. The six corrected estimates (summarised in Section 3.5) do not converge on a common effect size or scale: RR values range from 0.15 to 0.94, while the rater-based and adversarial estimates are reported as odds ratios (1.42–6.69); only the matched-pairs design (Table 8) crosses the conventional significance threshold, in the reversed direction. What is consistent across all six is direction—none places the agent above the human rate—not magnitude. We therefore state the finding in two tiers. The robust, well-supported claim is that the naive 1.64-fold agent excess is an artefact and does not survive correction by any method we apply; most corrected estimates are statistically indistinguishable from no difference. The weaker, more tentative claim is that one specific, more tightly controlled analysis (Table 8) suggests human fixes may recidivate more often in this corpus once both the workflow artefact and task-allocation confounding are addressed; given its limited coverage and the multiple analytical choices available in its construction (matching window, distance metric), we present this as hypothesis-generating rather than confirmatory: coverage is limited, we did not pre-register a matching protocol before observing these data, and — more fundamentally — this specific estimate depends on an equal-recall assumption between groups that we have not validated and have a concrete mechanistic reason to doubt for the agent group (Section 8.1); we do not consider it established until that assumption is tested.

7.3 On responsible reporting

This case illustrates that statistical robustness—surviving repository clustering, holding across statistical tests—is not the same as measurement validity. All robustness checks operate downstream of the measurement; if the measurement is biased, the bias propagates through all of them. For mined constructs, precision against ground truth is the check that matters, and it is most critical when the groups being compared differ systematically in their authoring workflows.

8 Threats to Validity

8.1 Construct validity

Our central claims concern construct validity — the degree to which DRD, as an indicator, actually captures genuine defect recurrence rather than an unrelated phenomenon. Despite its importance, construct validity is often assumed rather than assessed in software engineering research (Ralph and Tempero 2018); we address it directly through 200 manually labelled defects with operationalised rubrics. The remaining construct concern is label reliability. Batch-1 agreement is substantial ($\kappa = 0.754$) but batch-2 is moderate ($\kappa = 0.563$), with a high UNCLEAR rate (35%) driven by genuinely ambiguous cases. We publish both batches with rationales for community audit. Critically, the reversal finding is consistent and strengthens across batches (batch-1 OR = 4.59, $p = 0.0061$; batch-2 OR = 11.79, $p = 0.0002$), and the UNCLEAR asymmetry in batch-2 (14 agent vs 21 human) is, if anything, conservative for our direction claim. The adversarial bound in Section 6.3, however, shows this strengthening pattern does not extend to the worst-case treatment of unresolved cases.

A second construct-validity consideration concerns recall rather than precision. Every validated quantity in Sections 5–6—precision by rater, the precision-corrected sensitivity bounds (Table 5), the adversarial bound (Table 6), and the programmatic Timeline-API correction (Table 7, Section 8.2)—is computed only on defects flagged as recidivist (DRD > 0), and none examines the DRD = 0 population (866 agent and 1,837 human defects) for false negatives. The corrected estimates therefore assume DRD's recall is comparable across groups; we note this assumption here and return to it as a concrete direction for future work in Section 8.6. This assumption requires a careful qualification that we did not state precisely in earlier drafts. A recall asymmetry affects the naive marginal comparison and the corrected marginal comparison (Table 7) symmetrically, since both are computed over the same flagged population; for those two estimates the consideration does not favour either direction of the argument over the naive claim it revises. It does not, however, leave the matched-pairs reversal (Table 8) untouched. That reversal depends specifically on the draft-to-final collapse operation, which acts only on flagged defects, so if agents' failed fixes disproportionately resurface as brand-new issue reports—plausible, since agent PR workflows often close the originating issue cleanly and a recurrence would then appear as an unlinked new issue that DRD's linked-PR definition cannot see—then DRD's recall is lower for the agent group, the corrected agent recidivism rate is under-counted, and the Table 8 reversal is biased in the agent-favourable direction by an amount we cannot presently bound. We therefore do not treat the recall-validation pass as an optional refinement: it is a precondition for the tentative Table 8 reversal being upgraded from suggestive to established, and we say so explicitly in Section 8.6.

One partial piece of recall evidence is already available, and we report it in Section 8.2: the draft-to-final collapse itself discards 5 of the 39 validated genuine recurrences (12.8%) — 2 agent and 3 human, all five concentrated in a single repository (dotnet/maui), the same repository flagged in Sections 5.2 and 8.3 as a source of measurement noise — so the correction is demonstrably not recall-neutral even within the flagged population. That measurement bounds the collapse operation only; it says nothing about false negatives among the DRD = 0 defects, which remain wholly untested.

8.2 Internal validity — the corrected protocol, executed at corpus scale

We executed the deduplication algorithm of Section 6.4 against the full corpus, using the GitHub Issues Timeline API (/repos/OWNER/REPO/issues/NUMBER/timeline) rather than the standard pull-request endpoint. A first attempt, querying only the current 'draft' field on the pull-request object, returned zero collapsible pairs: this field reflects only the current state of a PR, not its state at creation, so a PR authored as a draft and later merged appears as 'merged', never as 'draft', in that endpoint's response. We therefore queried the Issues Timeline endpoint directly for each multi-attempt defect with $A_d \geq 2$ in the mined corpus (151 in the Copilot-versus-human inference subset, plus one further defect from a non-Copilot agent outside the inference subset; ≈780 API calls total: one call per linked PR for basic metadata and one for its timeline). For each PR, we checked both the current 'draft' flag and historical 'converted_to_draft'/'ready_for_review' timeline events. A closed-without-merge fallback covered PRs abandoned without ever transitioning out of draft. This corrected implementation collapsed 154 draft-to-final pairs within the Copilot-versus-human inference subset (134 in the agent group, 20 in the human group) across 151 multi-attempt defects; the script also processed one additional multi-attempt defect from a non-Copilot agent (OpenAI Codex), outside the inference subset, which contributed one further collapse not counted here. One further caveat about this collapse count warrants explicit disclosure: it is not evenly distributed across defects. A single agent issue (Azure/autorest.typescript #3243, $A_d = 32$) accounts for 31 of the 134 agent collapses (23%) and 20% of all 154 collapses; this issue sits just below the $A_d \geq 50$ umbrella-issue exclusion threshold applied in Section 3.2. It does not, however, drive the corrected result: because all 32 of its linked PRs collapse to a single fix attempt ($DRD_{corrected} = 0$), removing this issue entirely leaves the corrected agent rate unchanged at 6.5% and the corrected RR unchanged at 0.88, since the issue is a non-recidivist under both the naive and corrected codings and contributes to neither group's recidivist count after correction. We flag it for transparency rather than because it alters any reported estimate. Applying these collapses to the full corpus reduces agent recidivism from the naive 13.5% to a corrected 6.5%, and human recidivism from 8.2% to 7.4%; the corrected relative risk is 0.88 ($\chi^2 = 0.69$, $p = 0.41$), no longer significant. This corpus-scale programmatic result is independent of the manual rater validation reported in Sections 5–6—it relies on structured GitHub timeline events rather than human reading of PR titles and comments—and it falls within the precision-corrected sensitivity bounds of Table 5 (RR range 0.15–0.94), providing convergent evidence from two independent methods that the naive OR of 1.74 does not survive correction. We note that the magnitude of the programmatic correction (RR ≈ 0.88) is somewhat more conservative than the central manual-validation-based estimate (RR ≈ 0.38, Table 5) and the primary-rater reversal (OR = 6.69 in the

opposite direction on the REAL/ARTEFACT scale); we attribute this difference to the programmatic algorithm targeting only the draft-to-final (Type-A) artefact specifically, leaving Type-B (non-bug task) artefacts uncorrected, whereas the manual precision estimates net out both artefact types simultaneously. The programmatic result should therefore be read as a conservative, mechanism-specific lower bound on the correction's effect, consistent in direction with, but smaller in magnitude than, the full manual-validation-based estimate.

Table 7
Programmatic, corpus-scale draft-to-final correction via the GitHub Issues Timeline API (151 multi-attempt defects in the Copilot-versus-human inference subset, ≈ 780 API calls; one further defect from a non-Copilot agent outside the inference subset was also processed but is excluded from this table)

Copilot PR agent (n = 1,001)	**Naive %**	**Corrected %***	**Pairs collapsed**	**RR**
	13.5%	**6.5%**	**134**	
Human (n = 2,002)	8.2%	7.4%	20	
Relative risk (agent/human)	**1.64**	**0.88**	—	**χ^2=0.69, p = 0.41**

** Corrected % = (recidivist defects − collapsed-pair defects newly at DRD = 0) / group size. The corrected relative risk is no longer significant (p = 41%), consistent with the manual-validation finding that the naive OR of 1.74 does not survive correction.*

The same validation sample also bounds the corrected metric's own precision, which we report because a corrected between-group comparison does not by itself establish a valid absolute measure. Joining the 200 manually validated defects to their DRD_corrected values, the algorithm de-flags 54 of the 159 classifiable defects: 49 artefacts, correctly, and 5 genuine recurrences, incorrectly. Precision on the validated sample therefore rises from 24.5% to 32.4% (34 of 105 remaining flagged defects; Wilson 95% CI [24.2%, 41.8%]), and the direction reversal persists under the corrected metric (agent 16.2%, 95% CI [7.7%, 31.1%]; human 41.2%, 95% CI [30.3%, 53.0%]), while 12.8% of validated genuine recurrences (5 of 39) are lost. Two conclusions follow, and we state both plainly. First, the correction removes the between-group bias but does not make DRD a strong absolute measure of recurrence: it targets Type-A draft-to-final artefacts specifically, and the larger Type-B (non-bug task) category — 64% of manually identified artefacts — survives it untouched. Under the full corrected protocol of Section 6.4 (draft-to-final collapse together with bug-label and bot filtering) precision on this sample would rise to approximately 75%, but we report that figure only as an upper bound, because it is obtained by removing the artefacts our raters coded as non-bug rather than by applying a label-based filter prospectively at mining time, and a real filter would not match rater judgement exactly. Second, the collapse operation is not recall-free: it over-collapses at a measurable rate, discarding roughly one genuine recurrence for every ten artefacts it correctly **removes**. We report this cost rather than leaving it as an untested assumption; it is the first direct, if partial, evidence on DRD's recall in this study, and the full recall-validation pass of Section 8.6 remains the priority extension.

One caveat is fundamental to both the programmatic correction here and the matched-pairs result in Section 8.3, and we state it explicitly rather than leaving it implicit in the coding decision defended in Section 6.1. The draft-to-final correction is asymmetric by construction (134 agent collapses versus 20 human), and it is the single pivot of the corrected result: every corrected estimate that reverses or nullifies the naive signal does so because these agent collapses are treated as workflow artefacts rather than as genuine failed fix attempts. Section 6.1 defends the reading that a superseded draft PR is part of one fix attempt, and Section 5.5 reports a manual-precision sensitivity to the alternative reading (if draft-to-final pairs are instead counted as recidivism, agent precision rises to 26–31%, still below the human 41.3%). That alternative-reading sensitivity was, however, reported only for the manual precision estimate; it is equally applicable to the programmatic Table 7 and matched-pairs Table 8 results, and we therefore report both under an adversarial floor in which every agent draft-to-final collapse is credited back as a genuine failed attempt (the maximally agent-adverse reading, the mirror image of the collapse assumption). Under this floor, the programmatic corrected signal in Table 7 reverts essentially to the naive value: agent recidivism returns to 13.5% against human 7.4% (RR = 1.82, $\chi^2 = 28.3$, $p < 0.0001$). Even a half-floor, crediting back half of the agent collapses, leaves a significant agent excess (RR = 1.35, $p = 0.018$). The matched-pairs result in Table 8 is more resilient but not immune: under the same floor the corrected reversal disappears (agent-only discordant pairs rise from 6 to 25, RR = 1.57), yet the McNemar test does not reach significance in the agent-excess direction either ($p = 0.073$, n.s.), so the floor erases the reversal without establishing its opposite. The interpretation is therefore explicit: if a substantial fraction of the agent's draft-to-final pairs are genuine failed attempts rather than workflow artefacts, the corrected results over-collapse the agent group and the naive agent-excess signal partially returns. The independent check on this is the manual construct validation itself—two raters read these specific pairs and classified them as near-duplicate draft/final artefacts rather than distinct attempts, and even their most conservative alternative reading keeps agent precision below human precision—but that check rests on human judgement, which is exactly why we present the corrected reversal as robust to method choice yet contingent on the draft-to-final coding decision, and why the recall-validation and blinded-rater extensions of Section 8.6 remain the priority before the reversal is treated as established.

8.3 A second, distinct threat: task-allocation confounding, addressed via matched pairs

Beyond measurement artefacts, a substantive confound may affect any agent–human comparison drawn from this corpus. If Copilot's PR agent is systematically assigned different kinds of work than human contributors—for instance, more refactoring or documentation tasks, as the Type-B asymmetry in Table 4 (48 vs 29 non-bug-task artefacts) suggests—then any between-group difference, corrected or naive, partly reflects task allocation rather than fix durability. The human control set was constructed from the same 66 repositories as the agent set (Section 3.2), but the naive and corrected analyses in Sections 4–6 do not match individual defects on repository, time period, or complexity. We address this with an explicit matched-pairs design: for each agent defect, we identify the nearest-in-time human defect from the same repository within a 90-day window, yielding 179 matched pairs (17.9% of agent defects; coverage is limited because many repositories have too few human defects within the window

to match every agent defect, and we do not attempt unmatched comparisons). diff_additions/diff_deletions/diff_files were not populated during corpus construction (Section 3.2) and so complexity matching was not possible; the design therefore controls for repository and time period but not for fix complexity or issue type.

On this matched subset, the naive relative risk is attenuated relative to the full-corpus estimate (18.4% agent vs 12.3% human, RR = 1.50) and McNemar's exact test on the paired outcomes is not significant ($p = 0.108$), compared with $p = 0.002$ for the unmatched full-corpus GEE estimate. This indicates that part of the original naive significance was attributable to between-stratum composition (which repositories and time periods each group's defects happen to fall in) rather than to a within-stratum agent–human difference, independent of the draft-to-final mechanism. Applying the Timeline-API draft-to-final correction (Section 8.2) to this same matched subset sharpens the result further: corrected recidivism is 5.6% for the agent versus 11.7% for the matched human defects (RR = 0.48), and McNemar's exact test is now significant in the reversed direction ($p = 0.035$, Table 8). The reversal is directionally stable across matching windows: at 30, 60, 90, and 180 days the corrected relative risk is 0.75, 0.50, 0.48, and 0.52 respectively, with McNemar exact p-values of 0.61, 0.053, 0.035, and 0.035. It is significant at the 90-day (primary) and 180-day windows, marginal at 60 days ($p = 0.053$), and non-significant only at the narrowest 30-day window ($p = 0.61$, $n = 128$).

Two features of this matched subset warrant explicit disclosure, because the corrected significance in Table 8 rests on a small number of discordant pairs drawn from a small number of repositories. First, the 179 pairs come from only 16 repositories and are heavily concentrated: dotnet/maui alone contributes 52 pairs (29%), the top two repositories 43%, and the top four roughly 64%. Second, the corrected McNemar test is driven by just 23 discordant pairs (6 agent-only versus 17 human-only recidivist). This concentration matters especially because dotnet/maui is the same repository flagged in Section 5.2 as the source of shared-batch pull-request events that produced apparent rater inversions—so the repository most implicated in measurement noise is also the single largest contributor to the matched sample. We therefore test the reversal's robustness to repository concentration in two ways. (i) Leave-one-repository-out: recomputing the corrected McNemar test with each repository removed in turn, the reversal is not driven by dotnet/maui. Excluding it in fact strengthens the result (RR = 0.35, $p = 0.004$ on 19 discordant pairs). This is because dotnet/maui contributes only 4 discordant pairs and those lean marginally agent-adverse (3 agent-only versus 1 human-only). The result is, however, fragile to other single-repository exclusions: removing microsoft/vscode-pull-request-github (which contributes 6 human-only discordant pairs) drops the test to non-significance ($p = 0.21$), and several other single-repository removals push p just above 0.05. (ii) Cluster-robust sign test: aggregating to the repository level and testing the sign of each repository's net discordance, 8 repositories lean human-recidivist and 2 lean agent-recidivist, which is directionally consistent with the reversal but not significant under this fully clustered test ($p = 0.11$). We read these checks as showing that the direction of the matched-pairs reversal is stable—in particular, it is not an artefact of the noisiest repository—but that its statistical significance is not robust to repository clustering. This reinforces, rather than weakens, the paper's decision to present the Table 8 reversal as suggestive rather than confirmatory.

Table 8
Matched-pairs analysis: each agent defect paired with the nearest-in-time human defect from the same repository (90-day window, n = 179 pairs). McNemar's exact test on paired binary outcomes

Naive (matched, uncorrected)	Agent %	Human %	RR	McNemar p
	18.4%	12.3%	1.50	0.108 (n.s.)
Corrected (matched + Timeline-API)	5.6%	11.7%	0.48	0.035*

** Significant at α = 0.05, in the direction opposite to the naive agent-excess claim. Robust across matching windows of 60–180 days (p = 0.035–0.053); not significant at a 30-day window (p = 0.607, n = 128).*

This matched-pairs result is the most methodologically conservative estimate in the paper: it simultaneously controls for repository and time-period confounding (addressing the task-allocation concern raised in this section) and for the draft-to-final workflow artefact (Section 8.2), using a design with no rater judgement involved. It covers only 17.9% of agent defects, so it should not be read as superseding the full-corpus estimates in Table 7; rather, it corroborates them on the subset where the strongest available controls can be applied, and it is the strongest available evidence against the hypothesis that the corrected null result in Table 7 is itself an artefact of uncontrolled task allocation. The unmatched 82.1% of agent defects — typically in repositories with too few human-authored defects in the same time window to support matching — remain a source of residual uncertainty; we do not extrapolate the matched-pairs estimate to this unmatched majority, and a repository- and issue-type-stratified analysis with explicit complexity controls (once diff size is collected during corpus construction) remains a priority extension. A terminological note on this table: the 'RR' column reports the ratio of the two marginal recidivism rates (agent rate ÷ human rate on the matched subset), whereas the accompanying p-value comes from McNemar's exact test, which is a conditional test on the discordant pairs alone. The two are not the same estimand—the marginal rate ratio can move without the discordant-pair balance moving, and vice versa—so we report the rate ratio only as a descriptive effect-size summary and treat the McNemar p-value as the inferential quantity for the paired design.

8.4 External validity

The corpus is dominated by GitHub Copilot's coding agent on public repositories with more than one hundred stars from a single AIDev snapshot. The draft-then-final mechanism is a general property of how many coding agents author pull requests, so the qualitative caution should generalise. The 200-defect sample yields tighter precision intervals than prior single-batch studies, but a larger labelled corpus would further reduce uncertainty on group-level estimates. A further limitation stems from the corpus construction itself (Section 3.2): only pull requests that close an issue through an explicit GitHub closing reference are included, excluding fixes resolved without this syntax (e.g., closed manually, or referenced only in a commit message) from both groups — a selection bias, distinct from the artefacts in Section 6, that we have not quantified and that may not affect agents and humans equally.

8.5 Conclusion validity

Wilson CIs are used for precision estimates appropriate to small proportions. All chi-square statistics reported for 2×2 group comparisons (Sections 4, 8.2) apply the Yates continuity correction; the naive χ^2 = 19.84, the multiple-attempts χ^2 = 24.90, and the corrected χ^2 = 0.69 are the continuity-corrected values, and a reader recomputing without the correction will obtain slightly larger statistics (20.41, 25.79, and 0.82 respectively) that do not change any significance conclusion. Fisher's exact test and McNemar's exact test are used for direction tests appropriate to the cell counts and paired design respectively. Batch-level results are reported separately to demonstrate cross-batch consistency. We caution against over-interpreting the number of corrective analyses applied here as cumulative evidence of a uniform magnitude: with six corrected estimates, two rater choices, three artefact-resolution scenarios, and four matching-window choices for the matched-pairs design, the space of analytical decisions is large, and we did not pre-register a specific pipeline before observing the data (this study is explicitly exploratory, Section 3.4). We mitigate the resulting risk of selective reporting in four ways: we report every scenario we computed for the sensitivity analysis (Table 5) and every matching window we tested (Section 8.3), rather than only the most favourable; we report the adversarial bound (Table 6) as the least favourable case alongside the more favourable estimates; and we keep the headline robust claim narrow (the naive OR does not survive correction) rather than asserting the specific reversed magnitude as confirmed; and we disclose that the matched-pairs reversal (Table 8) rests on 23 discordant pairs concentrated in a few repositories, report leave-one-repository-out and cluster-robust checks showing its direction is stable but its significance is not robust to repository clustering (Section 8.3), and separately bound both the programmatic and matched-pairs corrections under an adversarial draft-to-final floor that partially restores the naive signal (Section 8.2). We do not over-claim the reversed direction in Table 8 as a confirmed durability verdict, and we note that both raters in the manual validation are co-authors with a stake in the reversal finding; we mitigate this via publication of all verdicts and rationales, computation of independent inter-rater reliability, and an explicit defence of the central coding decision (Section 6.1).

8.6 Future work

The priority extension is a recall-validation pass complementing the precision validation reported here. Because every corrected estimate in this paper is computed on the flagged (DRD > 0) population, none tests whether DRD misses genuine recurrences among defects it scores as clean (DRD = 0)—for example, a failed fix whose recurrence surfaces as a brand-new issue report rather than a reopen or a second linked pull request on the original issue, which DRD's linked-PR definition would not capture. To enable this check we constructed a stratified sample of 80 DRD = 0 defects (40 agent, 40 human), included unlabelled in the replication package together with a labelling protocol under which independent raters inspect each sampled defect for subsequent recurrence, including a keyword search for related follow-up issues. Labelling this sample would estimate DRD's recall separately by group and establish whether the equal-recall assumption underlying the corrected comparisons holds. We are deliberate about its status: as noted in Section 8.1, the tentative matched-pairs reversal in Table 8 rests on the flagged-only collapse operation and is therefore vulnerable to a group-asymmetric recall deficit in a way the robust marginal claim is not. We accordingly regard this recall pass not as a general improvement but as a specific precondition that must be satisfied before the Table 8 reversal is treated

as established rather than suggestive, and we caution readers against reading the corrected RR of 0.48 as a durability verdict until it is. It is the single most valuable next step for anyone building on this corpus, and Section 8.2 gives a first indication of why: the correction we do apply already loses 12.8% of validated genuine recurrences, so DRD's recall is measurably imperfect even before the unflagged population is examined. Two further extensions follow from limitations noted above: collecting diff-size fields during corpus construction would permit complexity-controlled matching in the matched-pairs design (Section 8.3), currently limited to repository and time period; and a blinded third-party labelling pass would remove the residual concern, noted in Section 8.5, that both raters in the present study are co-authors.

9 Conclusion

We set out to measure whether AI-assisted bug fixes stay fixed using an automatically mined recidivism metric, and found instead that the naive metric is not a valid estimator of relative fix durability. Defect Resolution Debt, applied naively to the 3,003-defect Copilot-versus-human inference subset (within a broader 3,040-defect mined corpus that also includes 37 sparsely represented non-Copilot agent defects), indicated a 1.64-fold agent excess surviving repository-clustered GEE regression (66 clusters). We draw a robust conclusion and a tentative one, and we keep them separate. The five robust estimates are consistent in direction — none places agent recidivism above human — though they vary in magnitude and significance: the recidivism-rate estimates (Rows 3–5) are statistically indistinguishable from no difference (RR 0.15–0.94, $p \geq 0.34$), while the precision estimates (Rows 1–2) show a significant reversal in precision. The tentative conclusion is that a more tightly controlled repository- and time-matched-pairs design (179 pairs, 17.9% coverage), which additionally addresses a task-allocation confound, finds a significant reversal after correction (RR = 0.48, McNemar $p = 0.035$). We present this as hypothesis-generating rather than confirmatory: it was not pre-registered, its coverage is limited, and it rests on an equal-recall assumption between groups that we have not validated and have concrete reason to doubt for the agent group (Section 8.1). Establishing it requires the recall-validation pass identified in Section 8.6. A further limitation applies to every corrected estimate in this paper: all rely on precision (false-positive control) validated against the flagged (DRD > 0) population, and none validates recall against the unflagged population; we constructed an 80-defect recall-check sample for this purpose and identify a recall-validation pass as the priority direction for future work (Section 8.6). Two artefact types drive the original naive inflation: agent-workflow artefacts (draft-to-final PR pairs, concentrated 134-to-20 in favour of the agent at corpus scale) and corpus-construction artefacts (non-bug task assignments). The latter is the larger category overall, at 64% of manually-identified artefacts; it affects both groups, though more often among agent defects (48 vs 29), and is left uncorrected by the programmatic algorithm, which targets Type-A artefacts specifically. We contribute a corrected measurement protocol for mined recidivism signals — not DRD itself, which we present as the specimen rather than the product — comprising a two-type validated artefact taxonomy, a precision-corrected sensitivity analysis with an adversarial robustness check, a deduplication algorithm validated both manually and programmatically at corpus scale, and a repository-and-time-matched-pairs analysis

addressing task-allocation confounding. We are explicit that the correction restores the validity of the between-group comparison without making DRD a strong absolute measure: its corrected precision on the validated sample is 32.4%, and the collapse operation itself discards 12.8% of validated genuine recurrences.

The general caution is methodological: a headline odds ratio computed from an unvalidated automated signal can overstate a real effect, understate it, or invert it, and the corrected magnitude can vary substantially depending on which validation method and which analytical choices are used; reporting a single corrected point estimate, rather than the range and the assumptions each method depends on, risks replacing one overclaim with another.

Declarations

Data Availability and Reproducibility

The mining pipeline, analysis scripts, figure-generation code, the derived dataset, and the complete two-batch validation data (both raters' verdicts and rationales for all 200 defects), together with the unlabelled 80-defect recall-check sample and its labelling protocol described in Section 8.6, are available at https://osf.io/9qmvy/overview?view_only=94d887886b6d4e399231a2a56f8763ec. The AIDev dataset is available at https://zenodo.org/records/16919272 (DOI 10.5281/zenodo.16919272); our fixed snapshot identifier is recorded in the replication package. The pipeline is deterministic and reproduces every figure and statistic from the recorded snapshot.

References

1. Asare O, Nagappan M, Asokan N (2023) Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code? Empirical Software Engineering 28(6). https://doi.org/10.1007/s10664-023-10380-1
2. da Costa DA, McIntosh S, Shang W, Kulesza U, Coelho R, Hassan AE (2017) A framework for evaluating the results of the SZZ approach for identifying bug-introducing changes. IEEE Transactions on Software Engineering 43(7):641-657. https://doi.org/10.1109/TSE.2016.2616306
3. Fu Y, Liang P, Tahir A, Li Z, Shahin M, Yu J, Chen J (2025) Security weaknesses of Copilot-generated code in GitHub projects: an empirical study. ACM Transactions on Software Engineering and Methodology 34(8):1-34. https://doi.org/10.1145/3716848
4. Jesse K, Ahmed T, Devanbu PT, Morgan E (2023) Large language models and simple, stupid bugs. In: 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia, pp 563-575. https://doi.org/10.1109/MSR59073.2023.00082
5. Jimenez CE, Yang J, Wettig A, Yao S, Pei K, Press O, Narasimhan K (2024) SWE-bench: can language models resolve real-world GitHub issues? In: 12th International Conference on Learning Representations (ICLR 2024), Vienna, Austria. https://openreview.net/pdf?id=VTF8yNQM66

6. Kalliamvakou E, Gousios G, Blincoe K, Singer L, German DM, Damian D (2016) An in-depth study of the promises and perils of mining GitHub. Empirical Software Engineering 21:2035-2071. https://doi.org/10.1007/s10664-015-9393-5
7. Kamei Y, Shihab E, Adams B, Hassan AE, Mockus A, Sinha A (2013) A large-scale empirical study of just-in-time quality assurance. IEEE Transactions on Software Engineering 39(6):757-773. https://doi.org/10.1109/TSE.2012.70
8. Keshavarz H, Nagappan M (2022) ApacheJIT: a large dataset for just-in-time defect prediction. In: MSR '22: Proceedings of the 19th International Conference on Mining Software Repositories, Pittsburgh, Pennsylvania, pp 191-195. https://doi.org/10.1145/3524842.3527996
9. Kim S, Zimmermann T, Pan K, Whitehead E (2006) Automatic identification of bug-introducing changes. In: 21st IEEE/ACM International Conference on Automated Software Engineering (ASE'06), Tokyo, Japan, pp 81-90. https://doi.org/10.1109/ASE.2006.23
10. Li H, Zhang H, Hassan AE (2025) The rise of AI teammates in software engineering (SE) 3.0: how autonomous coding agents are reshaping software engineering. arXiv. https://doi.org/10.48550/arXiv.2507.15003
11. Neto EC, da Costa DA, Kulesza U (2018) The impact of refactoring changes on the SZZ algorithm: an empirical study. In: 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Campobasso, Italy, pp 380-390. https://doi.org/10.1109/SANER.2018.8330225
12. Ralph P, Tempero E (2018) Construct validity in software engineering research and software metrics. In: Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering, pp 13-23. https://doi.org/10.1145/3210459.3210461
13. Rani P, Petrulio F, Bacchelli A (2024) On refining the SZZ algorithm with bug discussion data. Empirical Software Engineering 29:115. https://doi.org/10.1007/s10664-024-10511-2
14. Shihab E, Ihara A, Kamei Y, Ibrahim WM, Ohira M, Adams B, et al (2013) Studying re-opened bugs in open source software. Empirical Software Engineering 18:1005-1042. https://doi.org/10.1007/s10664-012-9228-6
15. Śliwerski J, Zimmermann T, Zeller A (2005) When do changes induce fixes? ACM SIGSOFT Software Engineering Notes 30(4):1-5. https://doi.org/10.1145/1082983.1083147
16. Tagra A, Zhang H, Rajbahadur GK, Hassan AE (2022) Revisiting reopened bugs in open source software systems. Empirical Software Engineering 27:92. https://doi.org/10.1007/s10664-022-10133-6
17. Xia X, Lo D, Shihab E, Wang X, Zhou B (2015) Automatic, high accuracy prediction of reopened bugs. Automated Software Engineering 22:75-109. https://doi.org/10.1007/s10515-014-0162-2
18. Zimmermann T, Nagappan N, Guo PJ, Murphy B (2012) Characterizing and predicting which bugs get reopened. In: 2012 34th International Conference on Software Engineering (ICSE), Zurich, Switzerland, pp 1074-1083. https://doi.org/10.1109/ICSE.2012.6227112

Figures

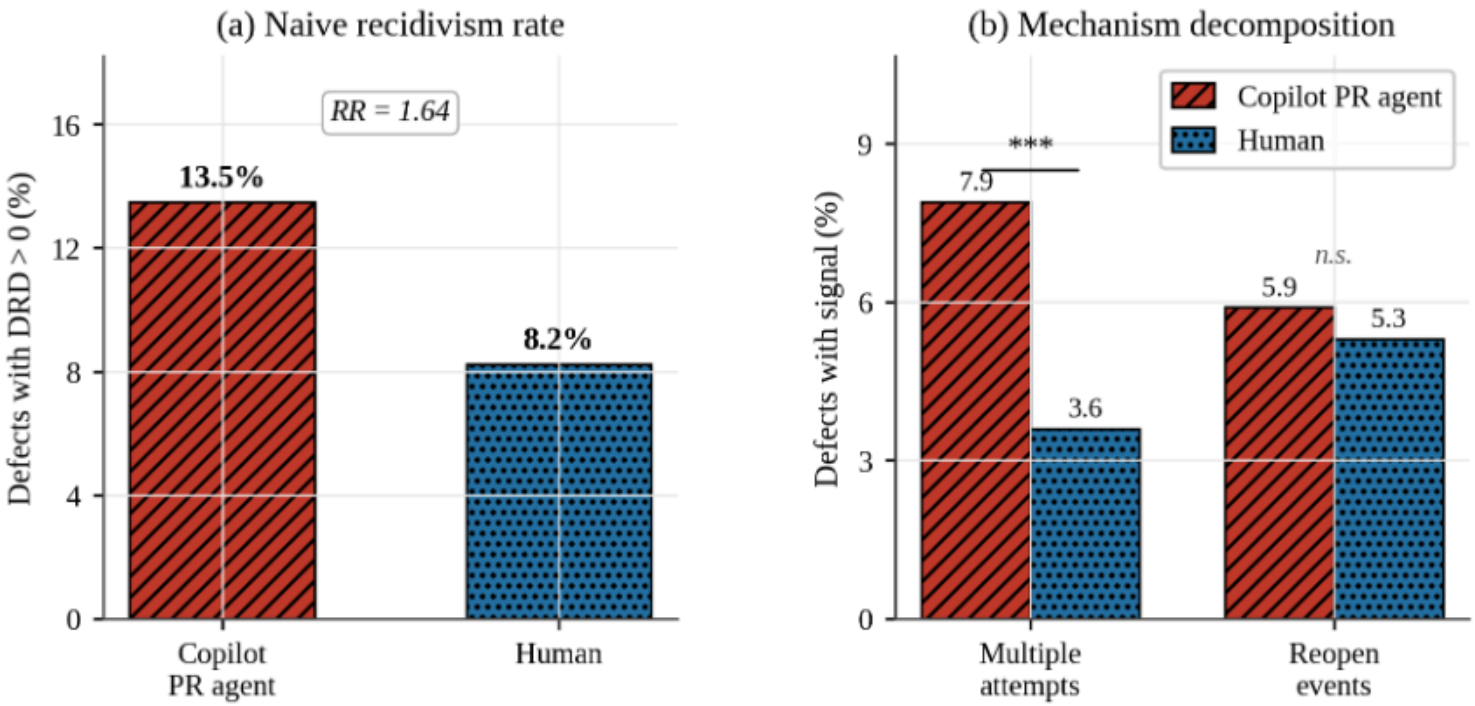


Figure 1

The naive automated recidivism signal. Panel (a): flagged recidivism rate (DRD > 0) is 1.64× higher for the Copilot agent than for humans. Panel (b): the gap is driven entirely by multiple fix attempts (***) and not by reopen events (n.s.)

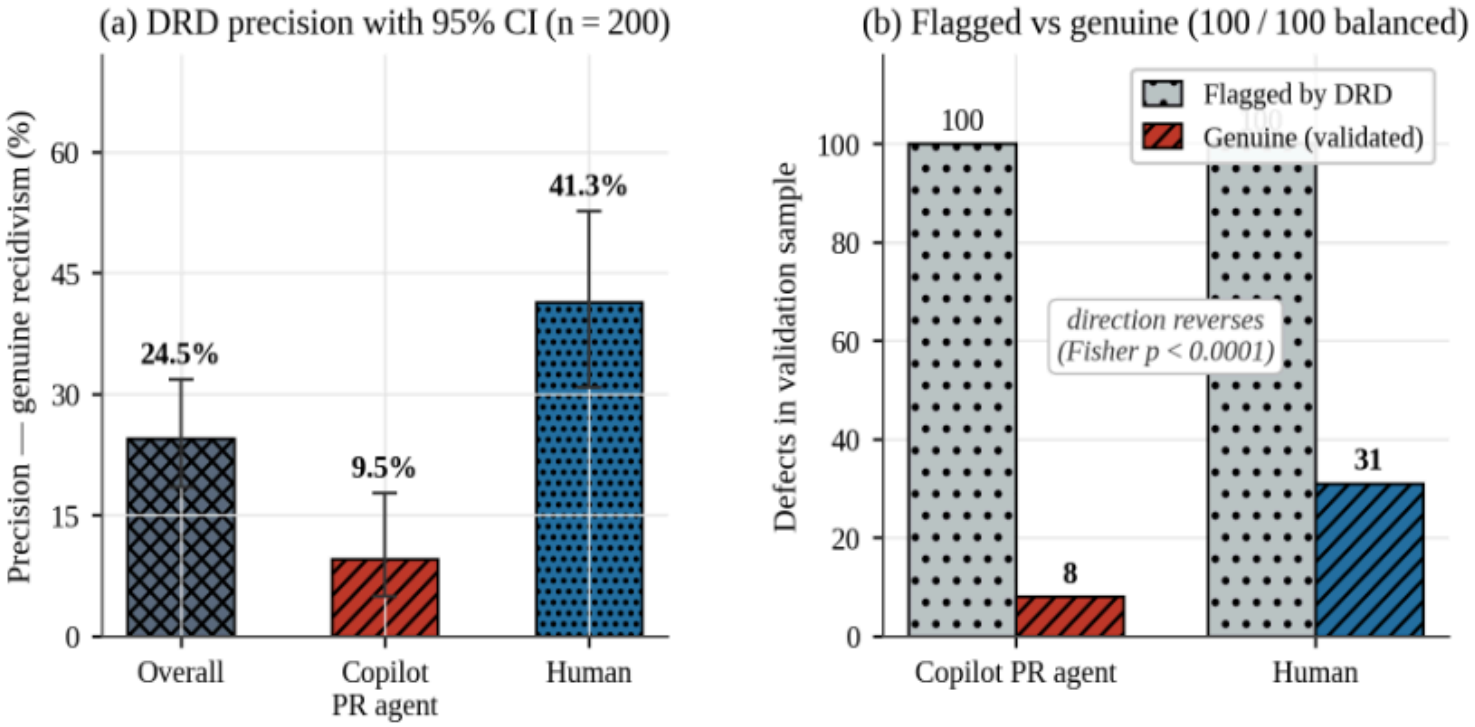


Figure 2

Construct validity of the automated signal (n=200 combined). Panel (a): DRD precision overall and by group with Wilson 95% CIs; agent 9.5% vs human 41.3%, non-overlapping intervals. Panel (b): both groups are flagged equally (100 each by design), but genuine recurrences number only 8 for the agent versus 31 for humans (Fisher's exact p <0.0001)

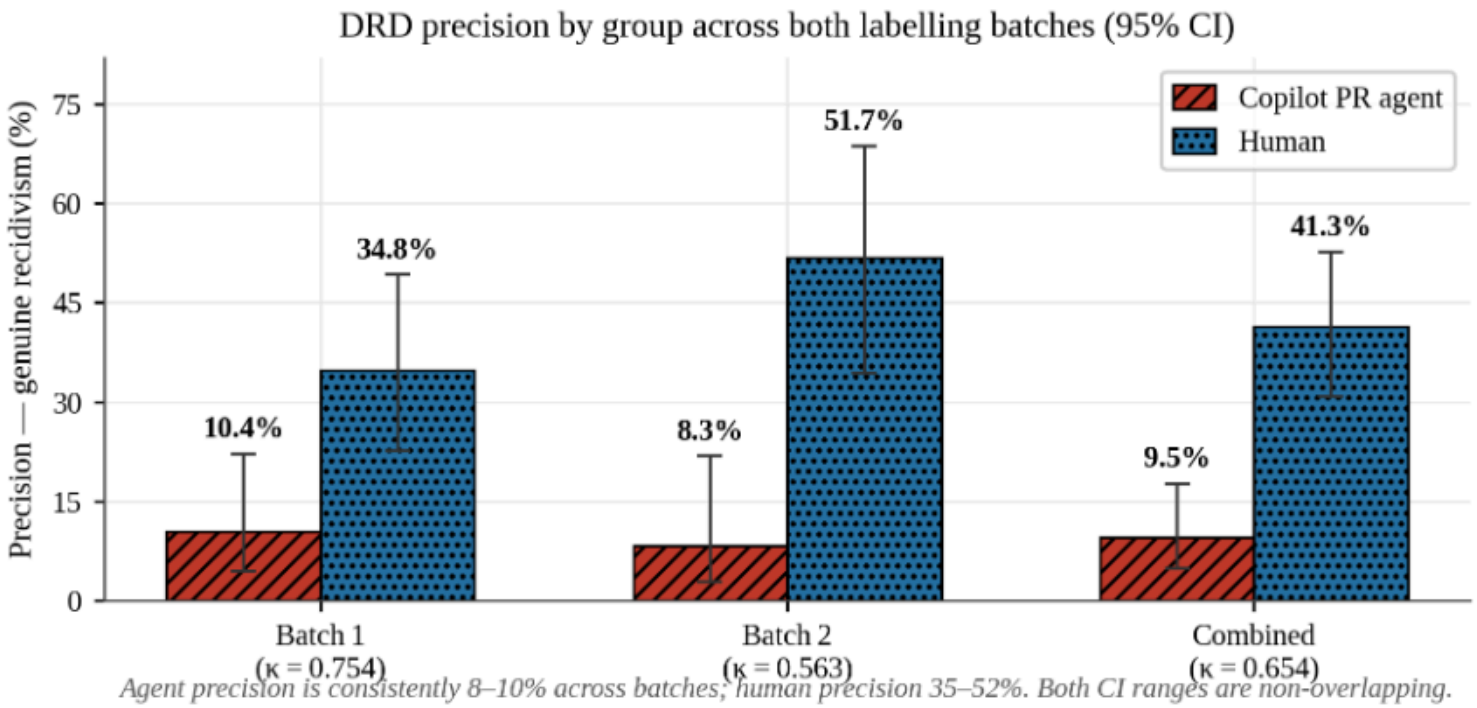


Figure 3

DRD precision by group across both labelling batches. Agent precision is consistently 8–10% across both batches; human precision is 35–52%. The directional differential is stable despite the lower κ in batch 2

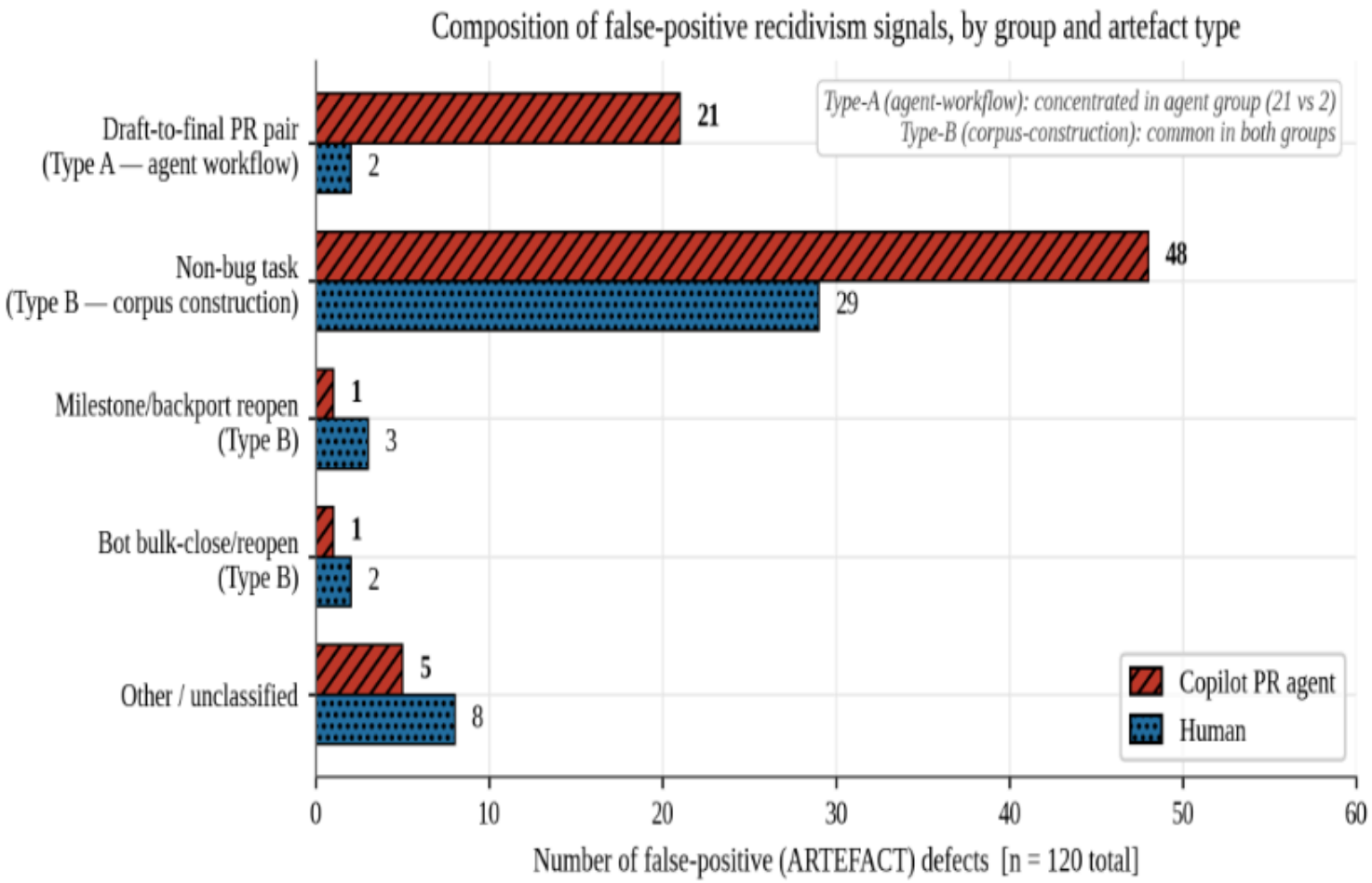


Figure 4

Composition of false-positive recidivism signals across both batches (120 artefacts). Type-A draft-to-final pairs (top) are asymmetric (17–21 agent vs 2 human) and explain the agent multiple-attempts gap. Type-B non-bug tasks (second) affect both groups and represent a corpus-construction issue.