montana *Research Foundation*

INSIGHT REPORT · SOFTWARE ENGINEERING

Coding Agents in Production

What controlled trials, delivery data, and the benchmarks themselves say about AI-assisted software engineering

REPORT

2026-02

Dr. Ricardo Arcifa, Francieli Carra
Montana Research Foundation

AUGUST 2026

EXECUTIVE SUMMARY

Engineering leaders are deciding how far to lean on coding agents on the strength of vendor demos and leaderboard scores. The published evidence is more mixed than either suggests: two controlled trials point in opposite directions, delivery telemetry shows stability falling as adoption rises, and the benchmark most often quoted has been retired by one of its own maintainers. This report sets out that evidence and the measurements a team should run on its own codebase before it changes how it works.

55.8%
Faster on a scoped task with Copilot in a 2023 controlled trial

19%
Slower on real issues in their own repositories, experienced developers, 2025 RCT

7.2%
Fall in delivery stability per 25% rise in AI adoption, DORA 2024

59.4%
Of audited SWE-bench Verified failures had flawed tests, per OpenAI

CONTENTS

METHOD & SCOPE

Sources are published studies and reports, each listed at the end: two controlled trials (Peng et al. 2023; METR 2025), the 2024 DORA report, GitClear's 2025 code-quality analysis of 211 million changed lines, the 2025 Stack Overflow Developer Survey, OpenAI's 2026 note retiring SWE-bench Verified, the SWE-Lancer paper, and Kapoor et al.'s AI Agents That Matter. Figures are quoted as published and were not re-run by the foundation.

Every chart in this report plots published numbers; none is illustrative. Where a source gives a range or a qualified figure, the qualification is kept. Recommendations are addressed to teams with an existing test suite and code review. Findings are stated for the studies, tools, and periods named, and not as general properties of coding agents.

KEY FINDINGS

01 The two best-known controlled trials disagree. Developers finished a scoped, greenfield task 55.8% faster with Copilot in 2023; experienced maintainers finished real issues in their own repositories 19% slower with early-2025 tools, while believing they were 20% faster.

02 At the organisation level, a 25% increase in AI adoption was associated with a 1.5% fall in delivery throughput and a 7.2% fall in delivery stability in the 2024 DORA survey, alongside gains in individual productivity and satisfaction.

03 Code quality telemetry moved the same way. Across 211 million changed lines, refactored code fell from 25% of changes in 2021 to under 10% in 2024, and duplicated blocks rose from 8.3% to 12.3%.

04 Developer trust has fallen as usage has risen. In 2025, 84% of developers used or planned to use AI tools, 46% distrusted their accuracy, and 66% said they spend more time fixing code that is almost right.

05 The benchmark most often quoted is no longer fit for the purpose. OpenAI found flawed tests in 59.4% of the SWE-bench Verified problems its model failed, plus contamination, and stopped reporting it in February 2026.

SECTION 01

01

Two trials, two answers

The strongest evidence on AI-assisted coding comes from randomised trials, and the two most cited ones reach opposite conclusions. They are both right, because they measure different work. Reading them together tells a team where the tools are likely to pay off and where they are likely to cost time.

The 2023 Copilot trial

Peng, Kalliamvakou, Cihon, and Demirer recruited developers and asked them to implement an HTTP server in JavaScript as fast as they could. The treatment group had GitHub Copilot; the control group did not. The treatment group finished 55.8% faster. The task was self-contained, well specified, and in a language and domain the models had seen many thousands of times. The authors also noted heterogeneous effects, with larger gains for less experienced developers.

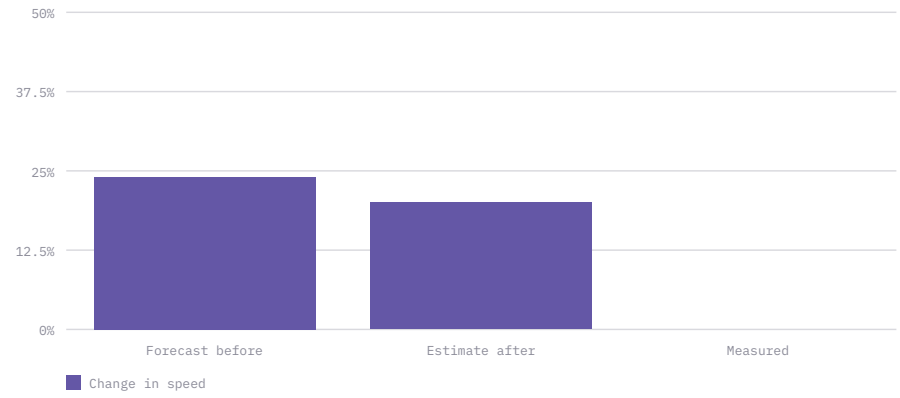
The 2025 METR trial

METR ran a randomised controlled trial with 16 experienced open-source developers working on 246 real issues in their own repositories, mostly large, mature projects. Issues were randomly assigned to allow or disallow AI tools, mainly Cursor Pro with Claude 3.5 and 3.7 Sonnet. With AI allowed, developers took 19% longer. Before the study they forecast a 24% speed-up; after it, they estimated 20%. The gap between perception and measurement is the finding that should worry anyone justifying a rollout on developer self-report.

19%

Slower with AI tools on real issues, against a self-estimate of 20% faster

METR 2025: expected speed-up against measured change in task time



Percentage change in time to complete real issues with AI tools allowed. Developers forecast a 24% speed-up, estimated 20% afterwards, and were measured at 19% slower (METR, July 2025; 16 developers, 246 issues).

Reading the two together

The Copilot trial measured a task that fits in one sitting and needs no knowledge of an existing system. The METR trial measured what senior engineers do most: changes to large codebases under constraints the model has never seen. The tools speed up bounded, well-specified work, slow down work whose difficulty is in the context, and developers cannot tell the two apart from the inside.

02

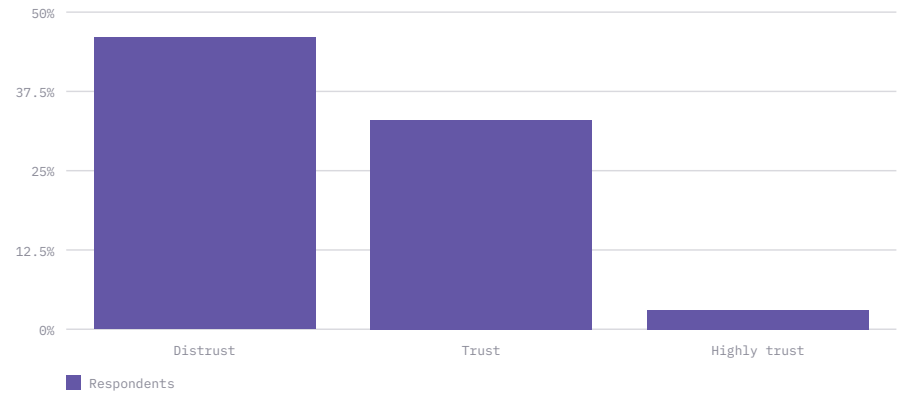
What ships

Individual speed is one measurement. What an engineering organisation delivers, and how often it breaks, is another. Two large datasets, one from surveys and one from repositories, report the same direction.

What developers say

The 2025 Stack Overflow Developer Survey put usage and trust side by side. 84% of respondents used or planned to use AI tools, up from 76% the year before. 46% actively distrusted the accuracy of those tools, 33% trusted it, and 3% highly trusted it. The most cited frustration, at 45%, was solutions that are almost right; 66% said they were spending more time fixing such code. Rising adoption alongside falling trust is the profile of a tool people are required to use and have learned to check.

Stack Overflow 2025: developers' trust in the accuracy of AI tools



Share of respondents by stated trust in AI tool accuracy, 2025 Stack Overflow Developer Survey. 84% of the same respondents used or planned to use AI tools.

Delivery throughput and stability

The 2024 DORA report, drawing on its annual survey of software teams, found that AI adoption raised individual productivity, flow, and job satisfaction, and that it lowered delivery performance. A 25% increase in AI adoption was associated with a 1.5% decrease in delivery throughput and a 7.2% decrease in delivery stability. DORA's explanation is consistent with its prior findings: AI makes it easier to produce more code per change, larger changes carry more risk, and the fundamentals of small batches and strong testing do not change because the code was generated.

What the repositories show

GitClear analysed 211 million changed lines of code from 2020 to 2024. Over that period, moved or refactored code fell from about 25% of changed lines in 2021 to under 10% in 2024, and duplicated code blocks rose from 8.3% to 12.3% of changed lines. In 2024, copy-pasted code exceeded moved code for the first time in the dataset. The mechanism is plain: a tool that writes new code on request lowers the cost of duplicating a block relative to finding and reusing the existing one.

7.2%

Fall in delivery stability associated with a 25% rise in AI adoption

03

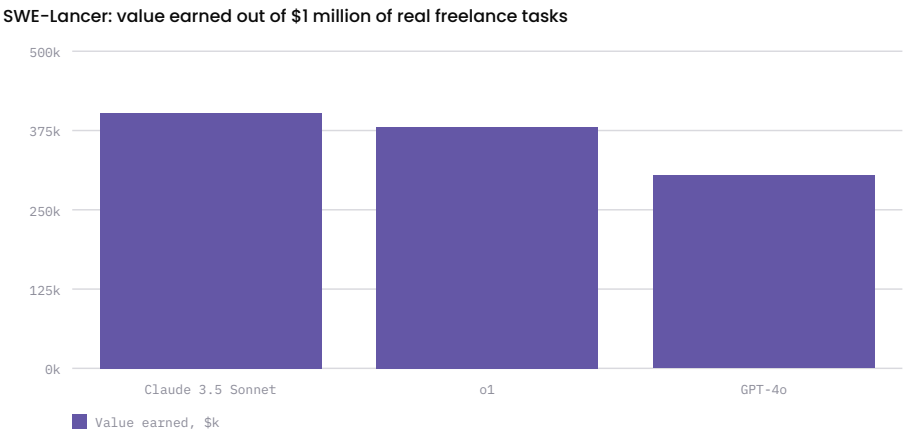
What the benchmarks measure

Vendor claims rest on benchmarks. One built from real freelance work gives a useful number and a low one; the one most often quoted was retired by one of its own maintainers this year.

SWE-Lancer prices the work

OpenAI's SWE-Lancer takes 1,488 freelance software tasks posted by Expensify on Upwork, with the prices Expensify actually paid, totalling $1 million. The 764 individual-contributor tasks, worth $414,775, ask the model to implement a fix or feature, checked by end-to-end browser tests. The 724 manager tasks, worth $585,225, ask it to choose the best of several proposals, graded against the one that was chosen. Claude 3.5 Sonnet, the best model in the paper, earned about $403,000; o1 earned about $380,000 and GPT-4o about $304,000. The authors' summary is that the majority of solutions are incorrect and that higher reliability is needed before trustworthy deployment.

A dollar-weighted score does two useful things. It weights tasks by what someone paid for them, and it shows that the unsolved 60% is where the money is: the expensive, multi-step tasks are the ones agents fail.



Dollar value of tasks solved by each model on the full SWE-Lancer set, in thousands of US dollars, as reported in the paper (arXiv 2502.12115). Total available: $1,000,000.

The retirement of SWE-bench Verified

SWE-bench Verified was a 500-task, human-screened subset of GitHub issues that became the standard coding number in model announcements. In February 2026, OpenAI said it would no longer evaluate models on it. The team audited the 138 problems its o3 model did not consistently solve across 64 runs and found material issues in test design or problem description in 59.4% of them: tests that checked the wrong behaviour, tests tied to details of the reference patch, and tests a correct fix could still fail. The audit also found signs that frontier models had seen the solutions during training, and recommended SWE-bench Pro instead.

59.4%

Of audited SWE-bench Verified failures had flawed tests or descriptions

04

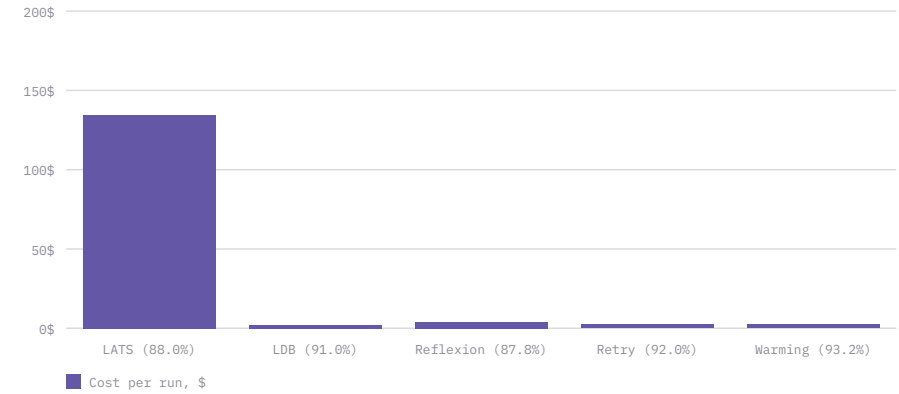
Cost is part of the score

Accuracy leaderboards leave out the cost of the run and, often, any holdout set. Kapoor, Narayanan, and colleagues showed both problems with numbers, and their finding changes which agent a team should choose.

Six scaffolds on HumanEval

On HumanEval, the authors compared published agent architectures with simple baselines. LATS with GPT-4 reached 88.0% accuracy at $134.50 for the run. LDB reached 91.0% for $2.19. Reflexion reached 87.8% for $3.90. Two baselines the authors built, retrying on failure and warming the sampling temperature, reached 92.0% for $2.51 and 93.2% for $2.45. The accuracies span five points; the costs span a factor of about 60. A leaderboard reporting accuracy alone would pick the most expensive scaffold.

HumanEval: cost of one evaluation run by agent architecture



Total cost in US dollars for five runs over HumanEval, from Kapoor et al., Table A1. Accuracy in brackets. LATS (GPT-4) 88.0%; LDB 91.0%; Reflexion 87.8%; Retry (GPT-4) 92.0%; Warming (GPT-4) 93.2%.

~60×

Cost spread between scaffolds within five accuracy points

> *Accuracy alone rewards the most expensive way to be right. Cost belongs on the leaderboard next to it.*
>
> AFTER KAPOOR ET AL., AI AGENTS THAT MATTER (2024)

Holdouts

The same paper surveyed 17 agent benchmarks and found that most lacked an appropriate held-out set. Seven had no holdout at all and no stated plan to add one. Among the eight domain-general benchmarks, one had an appropriate holdout. The authors give WebArena as an example, where an agent that hard-codes policies for the benchmark's tasks can top the leaderboard while telling a buyer nothing about unseen work.

05

Recommendations

The evidence supports a narrower rollout than the leaderboards imply and a wider one than the METR headline implies. The difference is closed by measurement the team controls: its own tasks, its own delivery metrics, and its own cost accounting.

FOR	ASK FOR	BECAUSE
Engineering leaders	A before-and-after on delivery throughput and change-failure rate instead of self-reported speed	Developers estimated +20% while measuring −19%
Platform teams	Batch-size limits and review load monitored during rollout	DORA ties AI adoption to larger changes and a 7.2% stability fall
Tech leads	Duplication and refactor ratios tracked per repository	Duplicated blocks rose from 8.3% to 12.3% of changes
Procurement	Vendor results on the team's own held-out tasks, with cost per task	SWE-bench Verified is contaminated; cost is absent from leaderboards
Finance	Cost per merged change, retries included	Scaffolds within five accuracy points differed ~60× in cost

> **The one-line version.** Measure delivery, never sentiment; pilot on bounded work first; keep a private task set; and put cost next to every accuracy number.

A checklist for the pilot

1. **Pick bounded work first.** Start where the 2023 trial found gains: well-specified tasks with little dependence on existing system context.
2. **Instrument delivery before the pilot.** Deployment frequency, lead time, change-failure rate, and time to restore, so the DORA effect is visible if it appears.
3. **Cap batch size.** Larger changes were DORA's explanation for the stability fall; enforce the limit in review.
4. **Track duplication.** Add a duplicated-block and refactor-ratio check to CI, per repository.
5. **Build a private task set.** Thirty to fifty tasks from closed issues, tests written after the fix, never shared with a vendor.
6. **Report cost with accuracy.** Cost per merged change, including retries, alongside pass rate on the private set.
7. **Ignore self-report.** Survey sentiment for morale, never for the productivity number.

The trials and datasets above cover particular tools, periods, and populations, and METR's authors list reasons their result may not generalise. A team's own measurements, taken before and after, are the only evidence that applies to that team.

Sources

1. Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot.* arXiv:2302.06590.
2. METR (2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity.* metr.org, 10 July 2025.
3. DORA (2024). *Accelerate State of DevOps Report 2024.* dora.dev.
4. GitClear (2025). *AI Copilot Code Quality: 2025 Data Suggests 4x Growth in Code Clones.* gitclear.com.
5. Stack Overflow (2025). *2025 Developer Survey: AI.* survey.stackoverflow.co/2025.
6. OpenAI (2026). *Why SWE-bench Verified no longer measures frontier coding capabilities.* openai.com, February 2026.
7. Miserendino, S., Wang, M., Patwardhan, T., Heidecke, J. (2025). *SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?* arXiv:2502.12115.
8. Kapoor, S., Stroebl, B., Siegel, Z., Nadgir, N., Narayanan, A. (2024). *AI Agents That Matter.* arXiv:2407.01502.

AUTHORS

Dr. Ricardo Arcifa
Montana Research Foundation

Francieli Carra
Montana Research Foundation

montana *Research Foundation*

An independent non-profit research institute doing open, rigorous research on the systems everyone is building. Every number in our work traces to a committed artifact, and every report cites the studies behind it.

montanaresearch.org

MRF-R-2026-02 · August 2026 ·