stonetusker

Stonetusker Insights · Leadership note on engineering delivery metrics

Faster Coding Is Not Faster Delivery: Measure What Your Coding Assistants Changed in 30 Minutes

The public studies on coding assistants are now more than a year old, and one has been revised by its own authors. The number that settles the budget discussion is your own. Most of it is already in your Git history.

For: Teams that use coding assistants, or plan to.

Skip if: You do not use coding assistants yet.

By Subeesh Sivanandan, Founder, Stonetusker Systems · Updated 29 October 2026 · About 4 minutes to read · PDF version

The short version

- The 2025 DevOps Research and Assessment (DORA) report links adoption of artificial intelligence to higher delivery throughput and to lower delivery stability.[1]
- Developers save time writing code and lose time elsewhere: in an Atlassian survey, 68% save more than 10 hours a week with artificial intelligence, while 50% lose 10 or more hours a week to friction.[3]

- The best known trial on speed has been revisited by its own authors, who say their newer data is an unreliable signal.[5]
- Your own before-and-after numbers give the answer. Two queries on your Git history produce the first two in 30 minutes.

About 90%
Technology professionals who use artificial intelligence at work[1]

68%
Developers who save more than 10 hours a week with artificial intelligence[3]

50%
Developers who lose 10 or more hours a week to friction[3]

Why it matters now

The 2025 DevOps Research and Assessment report, based on nearly 5,000 technology professionals, describes artificial intelligence as an amplifier. Teams with good platforms, reliable tests and small batches become stronger. Teams with slow reviews and manual releases get more of the same, faster.[1,2]

In 2025 a randomised trial by METR found that experienced open-source developers took 19% longer with these tools on their own repositories, while they believed they were 20% faster.[4] In February 2026 METR wrote that its newer data gives an unreliable signal, because more developers declined to work on study tasks without these tools.[5,6]

Delivery is a sequence of queues: review, continuous integration, tests, release approval and deployment. When pull requests arrive faster and review capacity stays the same, they wait longer. A measurement of your own system shows where the gain stops.

Check your delivery numbers in 30 minutes

Step 1. Pick the five numbers

Median time from pull request opened to merged, median time to first review, time from merge to production, change failure rate, and pipeline duration with rerun rate. Use

medians, since one pull request that stayed open for three months distorts an average. Compare the same quarter across two years to remove seasonal effects.

Step 2. Median hours from opened to merged

This uses the GitHub command-line tool and `jq` for one repository over a quarter.

```
gh pr list --repo ORG/REPO --state merged --limit 1000 \
  --search "merged:2026-07-01..2026-09-30" --json createdAt,mergedAt
\
  | jq '[.[] | select(.mergedAt != null and .createdAt != null)
        | ((.mergedAt | fromdateiso8601) - (.createdAt |
fromdateiso8601)) / 3600]
        | sort | .[length / 2 | floor]'
```

Run it again with `merged:2025-07-01..2025-09-30` for the same quarter last year.

Step 3. Median hours to first review

```
gh pr list --repo ORG/REPO --state merged --limit 1000 \
  --search "merged:2026-07-01..2026-09-30" --json createdAt,reviews
\
  | jq '[.[] | select(any(.reviews[]; .submittedAt != null))
        | ((.reviews | map(select(.submittedAt != null) |
.submittedAt | fromdateiso8601) | min)
           - (.createdAt | fromdateiso8601)) / 3600]
        | sort | .[length / 2 | floor]'
```

On GitLab, the merge requests interface returns `created_at` and `merged_at` for the same calculation:

```
curl -s --header "PRIVATE-TOKEN: $GITLAB_TOKEN" \

"https://gitlab.example.com/api/v4/projects/PROJECT_ID/merge_request
s?state=merged&updated_after=2026-07-01T00:00:00Z&per_page=100"
```

Step 4. Add release, failure and pipeline data

Take the time from merge to production from deployment events or release tags, the change failure rate from incidents, rollbacks and hotfixes per deployment, and pipeline duration and reruns from your continuous integration history.

What the result means

What you find	What it means	First move
More merged pull requests, and the time from opened to merged also rises	The gain from faster coding sits in review	Limit pull request size and agree a review time expectation
Time to first review rises faster than volume	Review capacity has not kept pace	Rotate a reviewer on duty and let automated checks handle formatting
Long or rising time from merge to production	Accepted work waits for release	Replace manual approvals that add waiting time without adding safety
Change failure rate rises with volume	The pattern the research describes: throughput up, stability down	Stabilise tests and keep each change small

The plan, quarter by quarter

1. **Rebuild the earlier picture.** Pull the five numbers for the same quarter last year and this year for your main repositories.
2. **Find where work waits.** The longest wait, in review, pipeline or release, is the constraint to address first.
3. **Fix one constraint.** Pick one change, such as smaller pull requests, a reviewer on duty or quarantined unstable tests, and run it for one quarter.
4. **Measure every quarter.** Report the same five numbers each quarter so that the trend guides decisions and budget.

What is still open

- The studies differ in size, method and population, so their numbers describe tendencies. Your own measurement is the reliable guide for your organisation.
- Tools change quickly, and METR could not repeat its 2025 trial with reliable results. Read its February 2026 update before you quote the 19%.[5,6]
- Deployment and incident data often sit in other systems, so the merge-to-production number may need a manual sample at first.
- The same research programme published a separate report on the return on investment of assisted development in April 2026, for teams that need to defend a budget.[7]

A note to forward to your team

Coding assistants and delivery: our numbers

Research links artificial intelligence adoption to higher throughput and lower stability. We want to know where our gain goes.

Please run the queries in this note on [repositories] for July to September 2025 and July to September 2026, and reply by [date] with five numbers for each quarter:
1. Median hours from pull request opened to merged
2. Median hours to first review
3. Time from merge to production
4. Change failure rate
5. Pipeline duration and rerun rate

Owner: [name]
Note: https://stonetusker.com/insights/ai-delivery-baseline.html

Questions

Do coding assistants make software delivery faster?

They can. The 2025 DevOps Research and Assessment report links adoption to higher throughput and also to lower delivery stability. The outcome depends on review, testing, platform quality and release practices.

Which numbers should we track?

Median time from pull request opened to merged, median time to first review, time from merge to production, change failure rate, and pipeline duration with rerun rate.

Can we rebuild the earlier picture today?

Mostly. Pull request timestamps, reviews and merges remain in GitHub or GitLab history, and pipeline systems keep run history, so last year's baseline can be rebuilt today.

Is self-reported productivity reliable?

It is a useful signal alongside system data. In the 2025 METR trial, developers were 19% slower with these tools while they believed they were 20% faster. METR says its newer data on this question is an unreliable signal.

Want a second opinion?

Run the check yourself first. If you would like a second opinion, reply with your five numbers only, for the same quarter in 2025 and 2026. No repository names, code or command output are needed. Within five working days you receive a short written review showing where your work waits, the likely causes and the first change to try. It is free and confidential, and a call is optional.

Ask for a second opinion

The button opens an email to subeesh@stonetusker.com with the subject and the numbers to fill in. You can also use the contact page, or reply to the email or LinkedIn message that brought you here.

Next page for your situation

- 20-Minute Delivery Checkup: the same five numbers for any team.
- Model lineage and the Artificial Intelligence Act dates: if your product contains models.

Sources

1. Google Cloud: Announcing the 2025 DORA Report
2. InfoQ: DORA State of AI-assisted Software Development 2025
3. Atlassian: State of Developer Experience Report 2025
4. METR: Measuring the impact of early-2025 AI on experienced open-source developer productivity

5. METR: We are changing our developer productivity experiment design (February 2026)
6. TechCrunch: Coders are refusing to work without AI (May 2026)
7. DORA: The ROI of AI-assisted Software Development (April 2026)

About the author

Subeesh Sivanandan is the Founder and CEO of Stonetusker Systems. He has worked for more than 27 years in release engineering, DevOps, platform engineering & embedded Linux. His career includes serving as the Principal Engineer & Manager of Release Engineering at Stryker, alongside earlier roles at Nokia, VeriSign, and IP Infusion. LinkedIn profile.

Revision history (recent 3 updates)

Version 2.1, 29 October 2026. Every fact on this page is checked against a minimum of two sources each month.

Version	Date	Change
1.6	16 October 2026	Rewritten as an article for quick readers: introduction, summary of all major changes, what it means and first steps; one figure; commands folded for platform leads; print option and direct email offer
2.0	25 October 2026	Simpler layout and shorter text; the review offer now asks for a few counts only
2.1	29 October 2026	Who-this-is-for box, email offer with the numbers listed, and next pages