METR

We are Changing our Developer Productivity Experiment Design

CONTRIBUTORS

Joel Becker, Nate Rush, Tom Cunningham, David Rein, and Khalid Mahamud

DATE

February 24, 2026

METR previously published a paper which found the use of AI tools caused a 20% slowdown in completing tasks among experienced open-source developers, using data from February to June 2025.

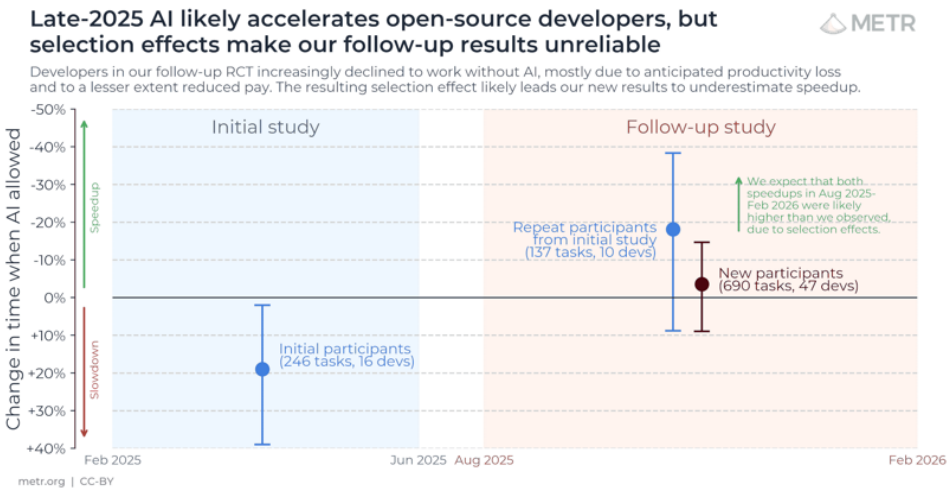
To understand how AI is impacting developer productivity over time, we started a new experiment in August 2025 with a larger pool of developers using the latest AI tools.

Unfortunately, given participant feedback and surveys, we believe that the data from our new experiment gives us an unreliable signal of the current productivity effect of AI tools. The primary reason is that we have observed a significant increase in developers choosing not to participate in the study because they do not wish to work without AI, which likely biases downwards our estimate of AI-assisted speedup. We additionally

believe there have been selection effects due to a lower pay rate (we reduced the pay from $150/hr to $50/hr), and that our measurements of time-spent on each task are unreliable for the fraction of developers who use multiple AI agents concurrently.

Based on conversations with study participants, we believe it is likely that developers are more sped up from AI tools now — in early 2026 — compared to our estimates from early 2025. However, because of the selection effects in our experiment, our data is only very weak evidence for the size of this increase.

Our raw results show some evidence for speedup. Our early 2025 study found the use of AI causes tasks to take 19% longer, with a confidence interval between +2% and +39%. For the subset of the original developers who participated in the later study, we now estimate a speedup of -18% with a confidence interval between -38% and +9%. Among newly-recruited developers the estimated speedup is -4%, with a confidence interval between -15% and +9%.



However the true speedup could be much higher among the developers and tasks which are selected out of the experiment. Some developers self-report very high speedups, though as we documented in our earlier study those estimates can be quite unreliable.

Due to the severity of these selection effects, we are working on changes to the design of our study. Below, we provide further detail and describe our plans for other means of studying the impact of AI on developer productivity.

Wider adoption of AI has made it more difficult to measure task-level productivity

Our second study, starting in August, consisted of 10 developers from the original study, plus a new set of 47 developers recruited from a more diverse set of open-source projects. The participants were paid $50/hour

for their participation.

As in the initial study, developers were asked to pre-specify each task that they intended to work on, and then submit the task-description for randomization. Each task was assigned to an “AI allowed” or “AI disallowed” condition. The developers would record the amount of time it took to complete the task, and we could thus compare the average time required to complete a typical task with and without AI.

Throughout 2025 there was an increase in the use of agentic tools among open-source developers, such as Claude Code and Codex. This wider adoption of AI has had two important effects in our study:

1. **Recruitment and retention of developers has become more difficult.** An increased share of developers say they would not want to do 50% of their work without AI, even though our study pays them $50/hour to work on tasks of their own choosing. Our study is thus systematically missing developers who have the most optimistic expectations about AI’s value.

2. **Developers have become more selective in which tasks they submit.** When surveyed, 30% to 50% of developers told us that they were choosing not to submit some tasks because they did not want to do them without AI. This implies we are systematically missing tasks which have high expected uplift from AI.

Together, these effects make it likely that our estimate

reported above is a lower-bound on the true productivity effects of AI on these developers. The selection effects seem to affect a minority share of developers and of tasks, which limits the degree of bias. However we are likely missing data on the most active adopters of AI, who may be of most interest. As AI capabilities continue to increase and developers' expectations grow as well, these effects will only get more dramatic, further limiting the validity of this study design.

Based on our interviews and surveys we believe the increase in selection is primarily due to higher expectation of AI-driven uplift by our developers. However the second study also had a lower pay rate, $50/hour instead of $150/hour in the original study, and this also likely contributed to the higher selection.

We also noticed some other problems, though we judged the magnitude of these to be less severe:

3. Some developers told us the types of tasks they attempted were different with agentic AI, leaning on the strengths of AI. This has two effects (a) the task-selection will be different between those in our study and those outside it; (b) the time-differences within the study may not represent value-differences, due to the substitution in task-type.

4. Some developers told us the quality of the final work, for a given task, was different between AI-allowed and AI-disallowed

conditions, e.g. differences in subjective code quality, or the amount of documentation or tests they chose to create.

5. Some developers were less likely to complete tasks that they submitted if they were assigned to the AI-disallowed condition. One developer did not complete any of the tasks that were assigned to the AI-disallowed condition.

6. Some developers reported it was challenging to report time-spent in completing tasks when they used agentic tools, because they would often work an unrelated task while waiting for the agent to complete its work.

Altogether, these issues make it challenging to interpret our central estimate, and we believe it is likely a bad proxy for the real productivity impact of AI tools on these developers.

Selected developer quotes

"I'm torn. I'd like to help provide updated data on this question but also I really like using AI!" — a developer from the original study early-2025 when asked to participate in the late-2025 study.

"I found I am actually heavily biased sampling the issues … I avoid issues like AI can finish things in just 2 hours, but I have to spend 20 hours. I will feel so painful if the task is decided as AI-disallowed." — a developer from the new study noting selection effects when

choosing what tasks to include in the study.

"my head's going to explode if I try to do too much the old fashioned way because it's like trying to get across the city walking when all of a sudden I was more used to taking an Uber." — a developer from the new study noting selection effects when choosing what tasks to include in the study.

Other means of measuring productivity

The impact of AI tools on developer productivity is an important input to AI R&D acceleration, and so we plan to continue exploring the following lines of research:

1. **More intensive experiments.** The selection issues would be alleviated if we had higher compliance, which could be achieved if we run shorter intense experiments, and possibly pay higher rates.

2. **Observational data.** There are many rich data sources on AI use in software development – from aggregate statistics (e.g. approximately 4% of GitHub commits are authored by Claude Code) to fine-grained transcripts (e.g. Sarkar (2025)). We intend to continue observation of our pool of open-source developers to understand how AI is used.

3. **Questionnaires.** Despite the difficulty of interpreting self-reported productivity

counterfactuals, and thus their potential biases, we are optimistic that careful choice of survey questions, along with time-use studies, could produce useful signals.

4. **Fixed-task experiments.** Our original study was novel in letting developers choose their own tasks, before randomization was applied. We could revert to a simpler design in which developers are given a fixed task to complete, either with or without AI assistance.

5. **Evaluations.** METR is continuing to build task evals, to measure the ability of agents to autonomously complete tasks, which has a clear relevance to productivity effects of agent adoption.

6. **Developer-level experiments.** An alternative design is to randomize at the developer level instead of the task level, i.e. each participant either uses AI for all their tasks or for none. This alleviates some of the task-level selection problems, however it makes the developer-level selection problems worse, and has lower power in detecting productivity effects.

As model capabilities continue to increase, we expect to continue to need to redesign some portions of our capability and measurement techniques.

Details of the productivity study

We paid developers $50/hour to work on their existing open-source projects, randomizing issues into working with AI-allowed or AI-disallowed. Developers were experienced open-source contributors with median 10 years experience. We have data from 57 developers, across 143 repos, and 800+ tasks.

10 of these developers are from the original study, and the remaining developers are new and recruited from a more diverse set of repositories—including smaller, more greenfield, and less mature repositories.

The full dataset from the early study is available here, and the most recent study here.