

The Photograph of a River

By Austen Tucker · 2026

The Photograph of a River

In July 2025, METR published the most rigorous study anyone had done on AI-assisted developer productivity. Sixteen experienced open-source developers. Two hundred forty-six tasks. Repos with millions of lines of code and tens of thousands of GitHub stars. Real work, real tools, real money, \$150 an hour.

The finding: developers using AI took 19% longer to complete their tasks.

That number ricocheted around the internet for months. AI skeptics waved it like a flag. AI boosters called the study flawed, the sample too small, the tools outdated. Everybody had an opinion. Almost nobody asked the question that mattered.

By the time you're arguing about the photograph, the river has moved.

Seven months later, METR changed their entire study design. Not because they got the first one wrong. Because the conditions that made the first study possible no longer existed.

Here's what happened. They tried to run a follow-up: same structure, randomized controlled trial, some developers use AI, some don't. And the "some don't" part fell apart.

Thirty to fifty percent of developers in the new cohort reported avoiding certain tasks because they didn't want to do them without AI. Not "preferred not to." Refused. Increasing numbers of developers wouldn't participate at all unless they could use their tools, even at fifty dollars an hour. One developer told the researchers: "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."

The randomized controlled trial, the gold standard of productivity research, was becoming structurally impossible. Not because of funding or methodology. Because the developers had changed.

This is the part nobody wants to talk about. Every AI productivity study is a snapshot of tools, habits, and workflows that are already different by the time the paper ships.

The METR study used Cursor Pro with Claude 3.5 and 3.7 Sonnet. Those were the best tools available in early 2025. By mid-2025 they were not. By the time anyone was citing the 19% slowdown in a strategy

meeting, the tools had gone through multiple generations, the prompting patterns had shifted, the agent scaffolding had improved, and the developers themselves had accumulated months of additional practice.

METR knew this. They said so in the original paper: "Future systems may overcome the challenges observed here through improvements in prompting techniques, agent scaffolding, or domain-specific fine tuning." They built their own expiration date into the findings. Almost nobody read that part.

The research lag problem is not unique to AI. It shows up everywhere that technology moves faster than academic publishing cycles. But in AI it's acute, because the tools don't iterate on a yearly schedule. They iterate on a monthly one. Sometimes weekly.

A drug trial that takes two years to publish is studying a molecule that hasn't changed. An AI productivity study that takes six months to publish is studying a tool that has been updated dozens of times. The molecule doesn't learn. The tool does.

So what are we actually measuring? Not the capability of AI coding tools. Not even the capability of developers using AI coding tools. We're measuring the capability of specific developers, using specific tools, at a specific moment, on specific tasks. By the time that measurement becomes a citation in someone's slide deck, every one of those variables has shifted.

This doesn't mean the research is useless. It means the research answers a different question than people think it does.

The METR study didn't prove that AI makes experienced developers slower. It proved that in early 2025, with early-2025 tools, experienced developers working on familiar codebases hadn't yet found workflows that made the tools net-positive for them. That's a finding about adoption curves and tool maturity and the gap between what experienced developers already know and what the tools can currently offer. It's important. But it's a photograph, not a prophecy.

The useful signal in these studies is structural, not numerical. The specific percentages expire. The patterns underneath them don't. Senior developers struggle more than juniors with current AI tools; that's structural. Code review burden increases when AI lowers the barrier to contribution; that's structural. Productivity gains plateau when only the coding phase gets faster; that's structural.

Those patterns will hold even as the numbers change. But the numbers are what get tweeted. The numbers are what get cited in board decks. The numbers are what people argue about at conferences while the river keeps moving underneath them.

Here's what I think is actually happening. We're trying to measure a transition with tools designed to measure a steady state.

The randomized controlled trial assumes you can cleanly separate "with AI" from "without AI." That assumption is dying. The developers who can't work without AI aren't outliers anymore; they're an increasing share of the population. The ones who can switch back and forth are increasingly the ones who haven't integrated the tools deeply enough for the integration to matter.

The best AI-assisted developers, the ones whose workflows have genuinely changed, are exactly the ones you can't study with a with/without design. They've rebuilt their process around the tools. Asking them to work without AI isn't a control condition. It's a different task.

METR saw this coming and pivoted to observational methods, surveys, and agent evaluations. They're not measuring "does AI help?" anymore. They're measuring "how is work changing?" That's a harder question. It's also the right one.

So the next time someone cites a number (AI makes developers 55% faster, or 19% slower, or exactly 10% more productive) ask when the study was done. Ask what tools they used. Ask whether the developers in the study had weeks of experience or months. Ask whether the ones who opted out were the best users or the worst.

And then remember that you're looking at a photograph of a river. The water was already somewhere else by the time they pressed the shutter.