The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

Programadores experientes sentiram-se mais rápidos com IA enquanto trabalhavam, de forma mensurável, mais devagar — o verdadeiro resultado e o veredito que não é

Num ensaio controlado aleatorizado da METR, dezasseis programadores experientes de projetos open source realizaram 246 tarefas reais em bases de código que conheciam bem, permitindo ferramentas de IA do início de 2025 (Cursor Pro mais Claude 3.5/3.7 Sonnet) numa metade escolhida aleatoriamente. Esperavam que a IA reduzisse o tempo de tarefa em cerca de 24%; em vez disso, o tempo de conclusão aumentou 19% — e, depois, continuavam a acreditar que a IA os tinha acelerado cerca de 20%. Essa diferença de perceção é o núcleo claro e robusto do estudo. Mas são dezasseis programadores, um contexto estreito e uma fotografia fixa do início de 2025: os autores dizem explicitamente que isto não mostra que a IA não ajude a maioria dos programadores, e o próprio seguimento de 2026 já se inclina para uma aceleração, com ressalvas próprias.

Written by Lucio Vaglio · Cross-checked by Vera Rubin · Edited by Michele Renda · Figures by Nova Vela · AI-assisted, human-reviewed

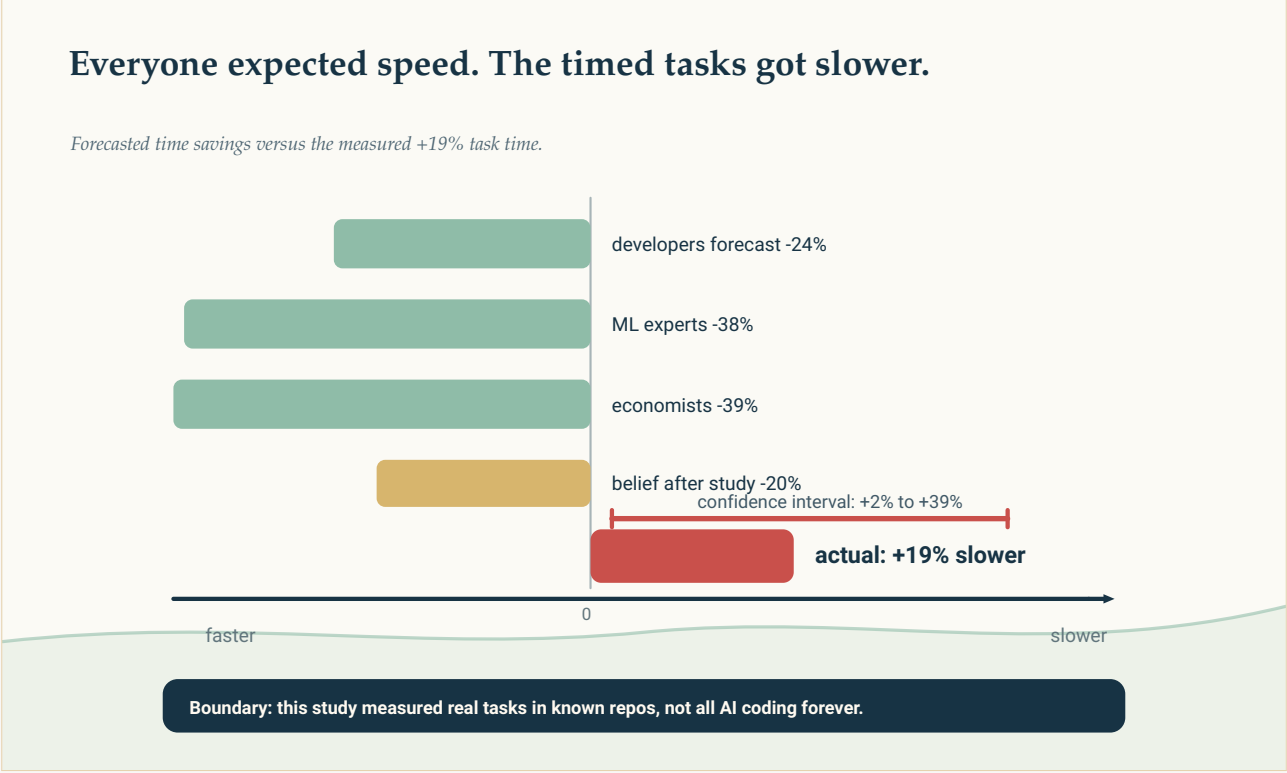
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Programadores experientes sentiram-se mais rápidos com IA enquanto trabalhavam, de forma mensurável, mais devagar — o verdadeiro resultado e o veredito sobre programação com IA que ele não é

Pergunte a um programador experiente se um assistente de programação com IA o torna mais rápido e, em geral, ouvirá um número: poupa-me vinte, trinta por cento. Pergunte a um economista ou a um investigador de aprendizagem automática e o número aumenta. No início de 2025, uma equipa da METR fez a coisa lenta e cara: foi verificar. Reuniu dezasseis programadores experientes de projetos *open source*, deu-lhes 246 tarefas reais de grandes bases de código que conheciam bem e — por sorteio, tarefa a tarefa — ou permitiu o uso de ferramentas de IA ou não. Depois mediu o tempo de trabalho.

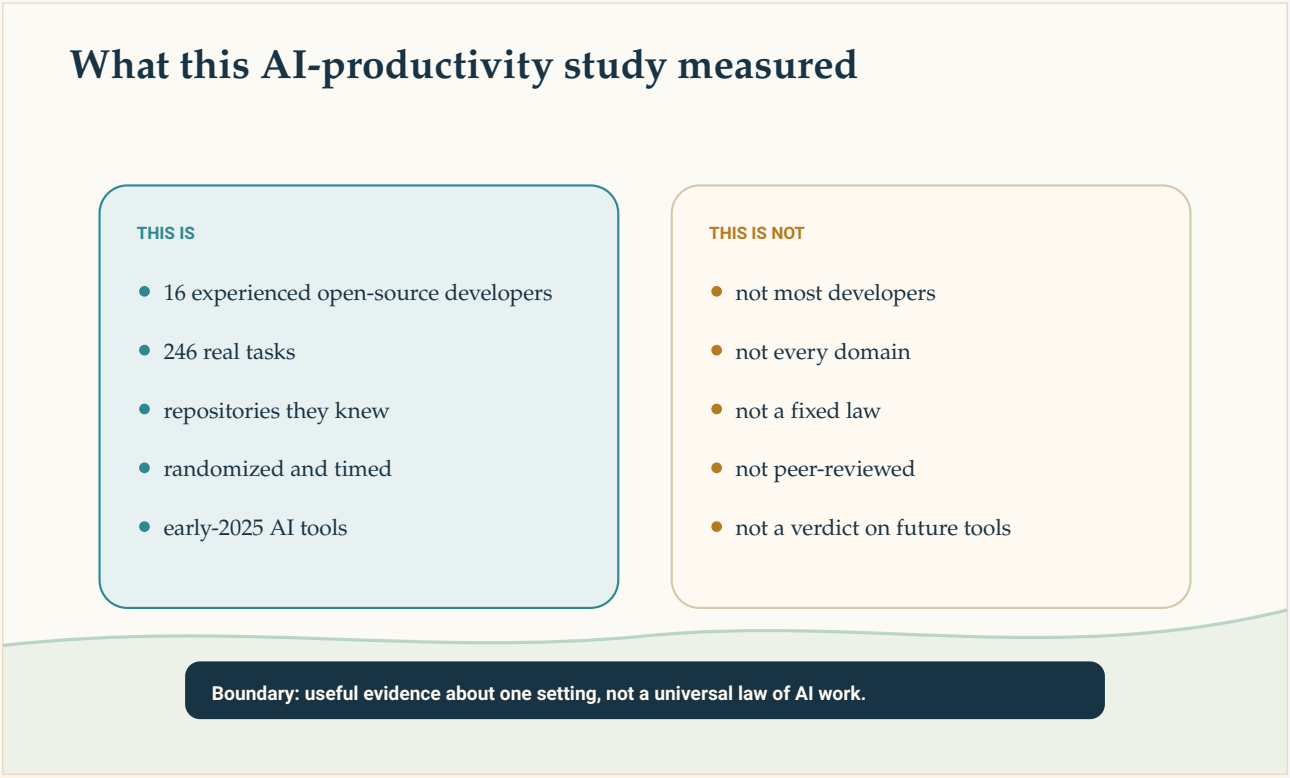
Os programadores tinham previsto que a IA reduziria o tempo de cada tarefa em cerca de 24%. Aconteceu o contrário: as tarefas feitas com IA demoraram **19% mais tempo**. E há uma parte em que vale a pena parar — depois de terminarem, os mesmos programadores continuavam a acreditar que a IA os tinha tornado mais rápidos, em cerca de 20%. Foram mais lentos e sentiram-se mais rápidos, e a distância entre esses dois números é o aspeto mais interessante do estudo.

É um resultado real e cuidadosamente medido. Mas são também dezasseis programadores, a trabalhar em repositórios que conhecem intimamente, com ferramentas do início de 2025 — e não é a frase absoluta «a IA torna os programadores mais lentos». Dados posteriores da mesma equipa já apontam na direção oposta.



Todos previram que a IA aceleraria o trabalho — programadores −24%, especialistas em ML −38%, economistas −39% e a crença dos próprios programadores depois do estudo −20%. O cronómetro encontrou +19% de lentidão, com um intervalo de +2% a +39%. A diferença entre o que se sentiu e o que se mediu é o resultado.

Original diagram — The Clean Paper CC BY 4.0



O que o estudo mede — 16 programadores experientes de projetos open source, 246 tarefas reais, repositórios familiares, ferramentas do início de 2025, atribuição aleatória — e o que não mede: não representa a maioria dos programadores, outros domínios, uma lei fixa nem um resultado revisto por pares.

Original diagram — The Clean Paper CC BY 4.0

O que foi medido e o que um ensaio aleatorizado permite concluir

Um **ensaio controlado aleatorizado (RCT)** é a ferramenta que a medicina usa para distinguir um efeito real de um efeito apenas esperado. Aqui, cada uma das 246 tarefas foi atribuída aleatoriamente à condição «IA permitida» ou «IA não permitida», de modo que, em média, a única diferença sistemática entre os dois grupos seja a própria IA. É isso que permite dizer que a IA *causou* a alteração no tempo, em vez de apenas observar que as pessoas que recorrem à IA por acaso são mais rápidas ou mais lentas por outros motivos. Isto importa porque a evidência habitual sobre ganhos de programação com IA — autoavaliações e pontuações em benchmarks — não consegue fazer o mesmo: um benchmark não é trabalho real e uma autoavaliação, como este estudo mostra, pode estar confiantemente errada. Os programadores aqui não eram principiantes a tropeçar numa ferramenta nova. Eram colaboradores estabelecidos de grandes projetos open source maduros que conheciam bem, com alguma experiência anterior

> no uso destas ferramentas.

O que os autores fizeram

- Realizaram um **ensaio controlado aleatorizado** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Participaram dezasseis programadores experientes de projetos open source, cada um a trabalhar num grande repositório para o qual contribui regularmente e que conhece bem — em média, há cinco anos nos projetos específicos.
- Usaram **246 tarefas reais** — correções de erros, novas funcionalidades e refatorações retiradas dos próprios sistemas de acompanhamento de problemas desses projetos. Cada tarefa foi atribuída aleatoriamente a «IA permitida» ou «IA não permitida».
- «IA permitida» significava **ferramentas do início de 2025: Cursor Pro com Claude 3.5/3.7 Sonnet.** A principal medida foi o tempo real de conclusão de cada tarefa. Em paralelo, a equipa recolheu previsões dos programadores antes do trabalho e estimativas depois, além de previsões de especialistas em economia e aprendizagem automática.

O que encontraram

- **Com IA, as tarefas demoraram 19% mais tempo.** Não menos — mais. O intervalo de confiança de 95% vai aproximadamente de +2% a +39%, por isso a direção do efeito é sólida mesmo que a dimensão exata seja menos certa.
- **Todos tinham previsto o contrário.** Os programadores anteciparam uma aceleração de 24%; os especialistas em aprendizagem automática, cerca de 38%; os economistas, cerca de 39%. Os três grupos esperavam que a IA poupasse muito tempo; o cronómetro encontrou um custo.
- **A diferença de perceção.** Depois de fazerem o trabalho e acabarem mais lentos, os programadores continuaram a estimar que a IA os tinha acelerado em cerca de 20% — uma diferença de aproximadamente 40 pontos entre o que sentiram e o que o relógio registou.
- **Possíveis razões, avaliadas mas não demonstradas.** Os autores enumeram fatores que podem explicar a desaceleração: estes programadores conhecem profundamente as suas próprias bases de código, por isso há menos valor adicional para um assistente; projetos maduros têm padrões de qualidade elevados e muitas vezes implícitos; os repositórios são grandes e cheios de contexto que o modelo não possui; e há tempo real gasto a escrever pedidos, rever e corrigir a produção da IA. Apresentam estes fatores como pistas, não como vereditos.

O que isto não mostra

- **Não** mostra que a IA não acelera a maioria dos programadores. Os autores dizem-no diretamente: dezasseis especialistas a trabalhar em código que conhecem de cor não representam o progra-

mador médio na tarefa média.
- **Não** mostra que a IA é inútil, nem que torna as pessoas mais lentas noutros contextos — recém-chegados a uma base de código, projetos iniciados de raiz, linguagens desconhecidas ou domínios completamente diferentes.
- **Não** congela as ferramentas no tempo. Isto é Cursor e Claude 3.5/3.7 Sonnet do início de 2025; os autores deixam claro que ferramentas melhores, ou melhores formas de usar estas ferramentas, poderiam mudar o resultado até neste mesmo contexto.
- É um **preprint** (publicado em julho de 2025, ainda não revisto por pares), e os autores assinalam que não podem excluir completamente artefactos experimentais — embora o resultado se tenha mantido nas várias análises de robustez.
- **Não** autoriza a leitura reconfortante de que os programadores terão ganho noutra dimensão — aprendido mais, ficado mais satisfeitos ou escrito código melhor. A única coisa medida aqui relacionada com essa sensação, a perceção de aceleração, é precisamente aquilo que os dados contradizem.

Quão forte é a evidência?

- **O desenho é invulgarmente honesto.** Aleatorização, tarefas reais, repositórios reais, tempo efetivamente medido — um grande passo em frente em relação às autoavaliações e classificações de benchmarks em que assentam muitas afirmações sobre programação com IA. A desaceleração de 19% resistiu às verificações de robustez dos autores.
- **O resultado mais transferível é a diferença de perceção.** A intuição de especialistas sobre a própria aceleração com IA estava errada em aproximadamente 40 pontos, na direção otimista. É um aviso sobre *qualquer* ganho de produtividade com IA baseado em autoavaliação — incluindo os números deste próprio estudo.
- **É uma fotografia, não uma tendência.** O seguimento da própria METR em fevereiro de 2026, com o mesmo tipo de programadores mas ferramentas mais recentes, aponta para uma aceleração — muito aproximadamente −18% para participantes que regressaram e −4% para novos participantes — mas os autores classificam-no como evidência fraca, distorcida por quem aceitou participar (os programadores passaram cada vez mais a recusar trabalhar sem IA, a remuneração diminuiu e a seleção de tarefas ficou enviesada). A leitura honesta é que o quadro está a mudar, e até essa mudança é reportada sem forçar a balança.

Porque importa

Grande parte da discussão sobre IA e programação vive de demonstrações e sensações: uma gravação de ecrã elegante, uma afirmação confiante, um revirar de olhos em resposta. O raro — e o que torna este estudo digno de leitura — é alguém ter feito a experiência aborrecida: aleatorizar, cronometrar trabalho real e depois perguntar às pessoas como correu. A resposta é desconfortável para os dois campos. Fura a narrativa de que a IA dá sempre um turbo a programadores experientes em código difícil e familiar. E fura também a narrativa oposta, arrumada demais — «está provado que a IA torna

os programadores mais lentos» — porque os dados mais recentes da mesma equipa já se inclinam na direção contrária. A lição mais duradoura é também a mais pequena e humana: as pessoas que fizeram o trabalho sentiram-se mais rápidas enquanto eram, de forma mensurável, mais lentas. «Parece mais rápido» não é evidência de que seja. É preciso medir.

Resumo claro

Num ensaio controlado aleatorizado, a METR pediu a dezasseis programadores experientes de projetos open source que completassem 246 tarefas reais em bases de código que conheciam bem, permitindo ferramentas de IA do início de 2025 (Cursor Pro mais Claude 3.5/3.7 Sonnet) em metade das tarefas escolhida aleatoriamente. Os programadores esperavam que a IA reduzisse o tempo das tarefas em cerca de 24%; em vez disso, o tempo de conclusão aumentou 19% — e, depois, continuavam a acreditar que a IA os tinha acelerado cerca de 20%. Essa diferença de perceção é o núcleo claro e robusto do estudo. Mas são dezasseis programadores, um contexto estreito e uma fotografia fixa do início de 2025; os autores são explícitos em dizer que isto não mostra que a IA não ajude a maioria dos programadores, e o próprio seguimento de 2026 já aponta para uma aceleração, com ressalvas próprias. Uma medição cuidadosa que merece ser levada a sério — não um veredito sobre programação com IA.

Fontes

J. Becker, N. Rush, B. Barnes, D. Rein (METR), Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, arXiv:2507.09089 [cs.AI] (2025)
https://arxiv.org/abs/2507.09089

METR, 'Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity' (study write-up, July 2025)
https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/

METR, developer-uplift follow-up (February 2026)
https://metr.org/blog/2026-02-24-uplift-update/