



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

Desenvolvedores experientes se sentiram mais rápidos com IA enquanto trabalhavam mensuravelmente mais devagar

Um ensaio controlado randomizado da METR fez dezesseis desenvolvedores experientes de código aberto executar 246 tarefas reais em codebases que conheciam bem, com ferramentas de IA do início de 2025 (Cursor Pro mais Claude 3.5/3.7 Sonnet) permitidas em uma metade aleatória. Eles esperavam que a IA reduzisse o tempo das tarefas em cerca de 24%; em vez disso, ela aumentou o tempo de conclusão em 19% — e depois eles ainda acreditavam que ela os havia acelerado em cerca de 20%. Essa lacuna de percepção é o núcleo mais forte do estudo. Mas são dezesseis desenvolvedores, um cenário estreito e um retrato fixo do início de 2025: os autores deixam claro que isso não mostra que a IA deixe de ajudar a maioria dos desenvolvedores, e o acompanhamento de 2026 da própria equipe já inclina para aceleração, com seus próprios caveats.

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

Desenvolvedores experientes se sentiram mais rápidos com IA enquanto trabalhavam mensuravelmente mais devagar — o achado real, e o veredito sobre programação com IA que ele não é

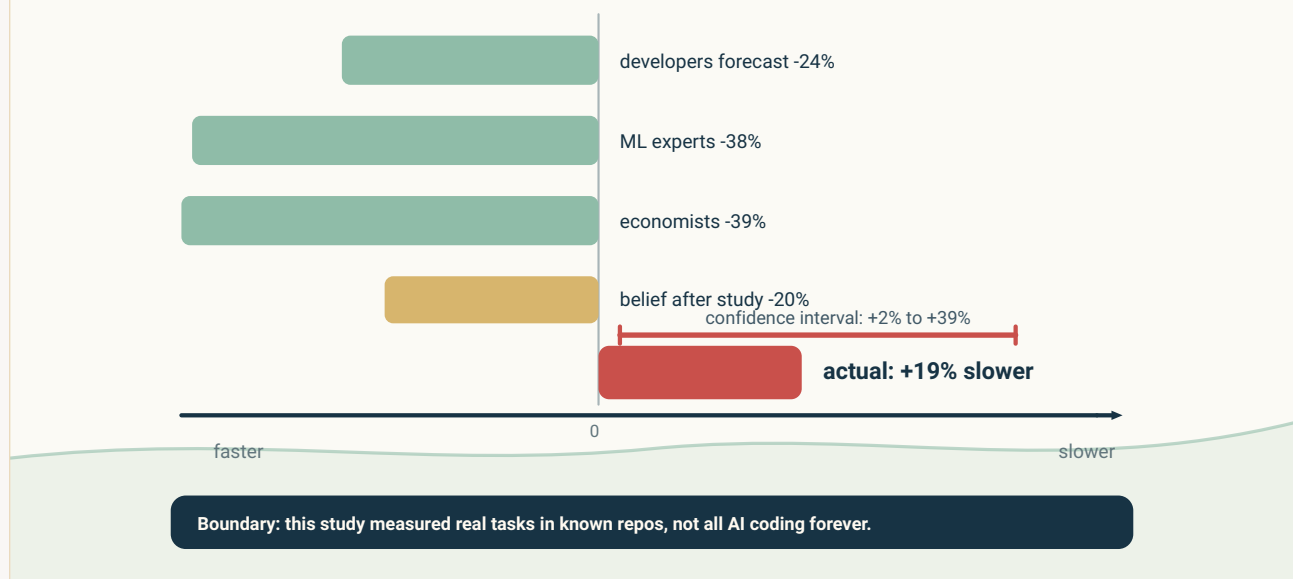
Pergunte a um programador experiente se um assistente de código com IA o acelera e você costuma ouvir um número: economiza vinte, trinta por cento. Pergunte a um economista, ou a um pesquisador de machine learning, e o número cresce. No início de 2025, uma equipe da METR fez a coisa lenta e cara: checkou. Pegou dezesseis desenvolvedores experientes de código aberto, entregou a eles 246 tarefas reais de grandes codebases que conheciam bem e — por sorteio, tarefa por tarefa — permitiu ou não o uso de ferramentas de IA. Depois cronometrou o trabalho.

Os desenvolvedores haviam previsto que a IA reduziria o tempo de tarefa em cerca de 24%. Aconteceu o contrário: as tarefas feitas com IA levaram **19% mais tempo**. E aqui está a parte que merece pausa: depois de terminar, os mesmos desenvolvedores ainda acreditavam que a IA os havia acelerado, em cerca de 20%. Eles foram mais lentos, sentiram-se mais rápidos, e a distância entre esses dois números é a coisa mais interessante do estudo.

Este é um resultado real e cuidadosamente medido. Também são dezesseis desenvolvedores, em repositórios que conhecem intimamente, usando ferramentas do início de 2025 — e não é a frase plana “IA deixa desenvolvedores mais lentos”. Dados posteriores da mesma equipe já apontam na outra direção.

Everyone expected speed. The timed tasks got slower.

Forecasted time savings versus the measured +19% task time.



Todos previram que a IA aceleraria o trabalho — desenvolvedores —24%, especialistas em machine learning —38%, economistas —39% e a crença posterior dos próprios desenvolvedores —20%. O cronômetro encontrou +19% mais lento, com intervalo de +2% a +39%. A lacuna entre o sentido e o medido é o achado.

Original diagram — The Clean Paper CC BY 4.0

What this AI-productivity study measured

THIS IS

- 16 experienced open-source developers
- 246 real tasks
- repositories they knew
- randomized and timed
- early-2025 AI tools

THIS IS NOT

- not most developers
- not every domain
- not a fixed law
- not peer-reviewed
- not a verdict on future tools

Boundary: useful evidence about one setting, not a universal law of AI work.

O que o estudo mede — 16 desenvolvedores experientes de código aberto, 246 tarefas reais, repositórios familiares, ferramentas do início de 2025, randomização — e o que ele não mede: não a maioria dos desenvolvedores, não outros domínios, não uma lei fixa, não revisão por pares.

Original diagram — The Clean Paper CC BY 4.0

O que foi medido, e o que um ensaio randomizado compra

Um **ensaio controlado randomizado (RCT)** é a ferramenta que a medicina usa para separar um efeito real de um esperado. Aqui, cada uma das 246 tarefas foi atribuída aleatoriamente a ser feita com IA permitida ou não, de modo que, em média, a única diferença sistemática entre os dois grupos é a própria IA. É isso que permite dizer que a IA *causou* a mudança no tempo, em vez de apenas observar que pessoas que recorrem à IA talvez sejam mais rápidas ou mais lentas por outros motivos. Isso importa porque a evidência usual para ganhos de programação com IA — autorrelatos e pontuações em benchmarks — não consegue fazer isso: um benchmark não é trabalho real, e um autorrelato, como este estudo mostra, pode estar confiante e errado. Os desenvolvedores aqui não eram novatos tropeçando num brinquedo novo. Eram contribuidores estabelecidos de projetos grandes e maduros que conheciam bem, com alguma experiência prévia no uso das ferramentas.

O que os autores fizeram

- Rodaram um **ensaio controlado randomizado** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Dezesseis desenvolvedores experientes de código aberto, cada um trabalhando em um grande repositório para o qual contribui regularmente e conhece bem — em média, cinco anos nos projetos específicos.
- Usaram **246 tarefas reais** — correções de bugs, funcionalidades e refatorações tiradas dos próprios issue trackers desses projetos. Cada tarefa foi aleatoriamente atribuída a “IA permitida” ou “IA proibida”.
- “IA permitida” significava **ferramentas do início de 2025: Cursor Pro com Claude 3.5/3.7 Sonnet**. A medida primária foi o tempo real de conclusão por tarefa. A equipe também coletou previsões dos desenvolvedores antes do trabalho, estimativas deles depois, e previsões de especialistas em economia e machine learning.

O que eles encontraram

- **Com IA, as tarefas levaram 19% mais tempo.** Não menos: mais. O intervalo de confiança de 95% vai de cerca de +2% a +39%, então a direção é sólida mesmo que o tamanho exato não seja.
- **Todo mundo havia previsto o oposto.** Desenvolvedores previram uma aceleração de 24%; especialistas em machine learning, cerca de 38%; economistas, cerca de 39%. Os três grupos esperavam que a IA economizasse muito tempo; o cronômetro encontrou custo.
- **A lacuna de percepção.** Depois de fazer o trabalho e sair mais lentos, os desenvolvedores ainda estimaram que a IA os havia acelerado em cerca de 20% — uma lacuna de aproximadamente 40 pontos entre o que sentiram e o que o relógio registrou.
- **Razões candidatas, ponderadas e não provadas.** Os autores listam fatores que poderiam explicar a desaceleração: esses desenvolvedores conhecem profundamente seus próprios codebases, então há menos para um assistente acrescentar; projetos maduros carregam padrões de qualidade altos e muitas vezes implícitos; os repositórios são grandes e cheios de contexto que o modelo não tem; e tempo real é gasto promptando, revisando e corrigindo saída de IA. Eles apresentam isso como pistas, não vereditos.

O que isso não mostra

- **Não** mostra que a IA falha em acelerar a maioria dos desenvolvedores. Os autores dizem isso diretamente: dezesseis especialistas em código que conhecem de cor não são o desenvolvedor médio na tarefa média.
- **Não** mostra que IA é inútil, ou que deixa pessoas mais lentas em outros cenários — recém-chegados a um codebase, trabalho greenfield, linguagens desconhecidas ou outros campos.
- **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 modos melhores de usar estas, poderiam

mudar o resultado mesmo nesse cenário exato.

- É um **preprint** (publicado em julho de 2025, ainda sem revisão por pares), e os autores observam que não conseguem descartar totalmente artefatos experimentais — embora o achado tenha sobrevivido às análises.
- **Não** autoriza a leitura reconfortante de que os desenvolvedores devem ter ganhado de outro jeito — aprendido mais, ficado mais felizes, escrito código melhor. A coisa medida aqui, a sensação de aceleração, é exatamente o que os dados contradizem.

Quão sólidas são as evidências?

- **O desenho é incomumente honesto.** Randomização, tarefas reais, repositórios reais, tempo real — um avanço grande em relação aos autorrelatos e rankings de benchmark nos quais repousa a maior parte das afirmações sobre programação com IA. A desaceleração de 19% sobreviveu às checagens de robustez dos autores.
- **O achado mais portátil é a lacuna de percepção.** A intuição de especialistas sobre o próprio ganho de velocidade com IA errou por cerca de 40 pontos, na direção otimista. Isso é um aviso sobre *todo* ganho de produtividade com IA relatado por autorrelato — inclusive os números deste estudo.
- **É um retrato, não uma tendência.** O acompanhamento de fevereiro de 2026 da própria METR, com o mesmo tipo de desenvolvedores em ferramentas mais novas, aponta para aceleração — muito aproximadamente — 18% para desenvolvedores que retornaram e —4% para novos — mas os autores o marcam como evidência fraca, distorcida por quem topou participar (desenvolvedores cada vez mais recusavam trabalhar sem IA, o pagamento caiu e a seleção de tarefas mudou). A leitura honesta é que o quadro está se movendo, e até esse movimento é relatado sem embelezamento.

Por que importa

A maior parte da discussão sobre IA e programação roda em demos e impressões: uma gravação de tela elegante, uma afirmação confiante, um revirar de olhos contrário. O raro, e o que torna isto digno de leitura, é que alguém fez o experimento sem glamour — randomizar, cronometrar trabalho real e depois perguntar como foi. A resposta é desconfortável para os dois lados. Fura a história de que IA turbinava uniformemente desenvolvedores experientes em código difícil e familiar. E fura o oposto arrumado — “IA deixa desenvolvedores mais lentos, provado” — porque os dados mais novos da mesma equipe já inclinam para o outro lado. A lição mais durável também é a menor e mais humana: as pessoas fazendo o trabalho se sentiram mais rápidas enquanto eram mensuravelmente mais lentas. “Parece mais rápido” não é evidência de que é.

Em resumo

Em um ensaio controlado randomizado, a METR fez dezesseis desenvolvedores experientes de código aberto concluir 246 tarefas reais em codebases que conheciam bem, com ferramentas de IA do início de 2025 (Cursor Pro mais Claude 3.5/3.7 Sonnet) permitidas em uma metade aleatória. Os desenvolvedores esperavam que a IA reduzisse o tempo de tarefa em cerca de 24%; em vez disso, ela aumentou o tempo de conclusão em 19% — e depois eles ainda acreditavam que ela os havia acelerado em cerca de 20%. Essa lacuna de percepção é o núcleo afiado e robusto do estudo. Mas são dezesseis desenvolvedores, um cenário estreito e um retrato fixo do início de 2025; os autores deixam claro que isso não mostra que a IA falhe em ajudar a maioria dos desenvolvedores, e o acompanhamento de 2026 da própria equipe já aponta para aceleração, com seus próprios caveats. 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/>