The Clean Paper

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

Lập trình viên giàu kinh nghiệm cảm thấy nhanh hơn với AI trong khi đo được là chậm hơn — phát hiện thật, và không phải phán quyết về AI coding

Một RCT của METR cho 16 lập trình viên open-source giàu kinh nghiệm làm 246 tác vụ thật trên codebase họ biết rõ, với công cụ AI đầu năm 2025 (Cursor Pro + Claude 3.5/3.7 Sonnet) được cho phép ngẫu nhiên ở một nửa. Họ kỳ vọng AI giảm thời gian khoảng 24%; thực tế thời gian hoàn thành tăng 19% — và sau đó vẫn tin AI tăng tốc mình khoảng 20%. Khoảng cách nhận thức là lõi sắc và vững của nghiên cứu. Nhưng đây là 16 lập trình viên, một bối cảnh hẹp và ảnh chụp đầu năm 2025: tác giả nói rõ nó không cho thấy AI không giúp phần lớn lập trình viên, và follow-up 2026 của chính họ đã nghiêng về tăng tốc, với lưu ý riêng.

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

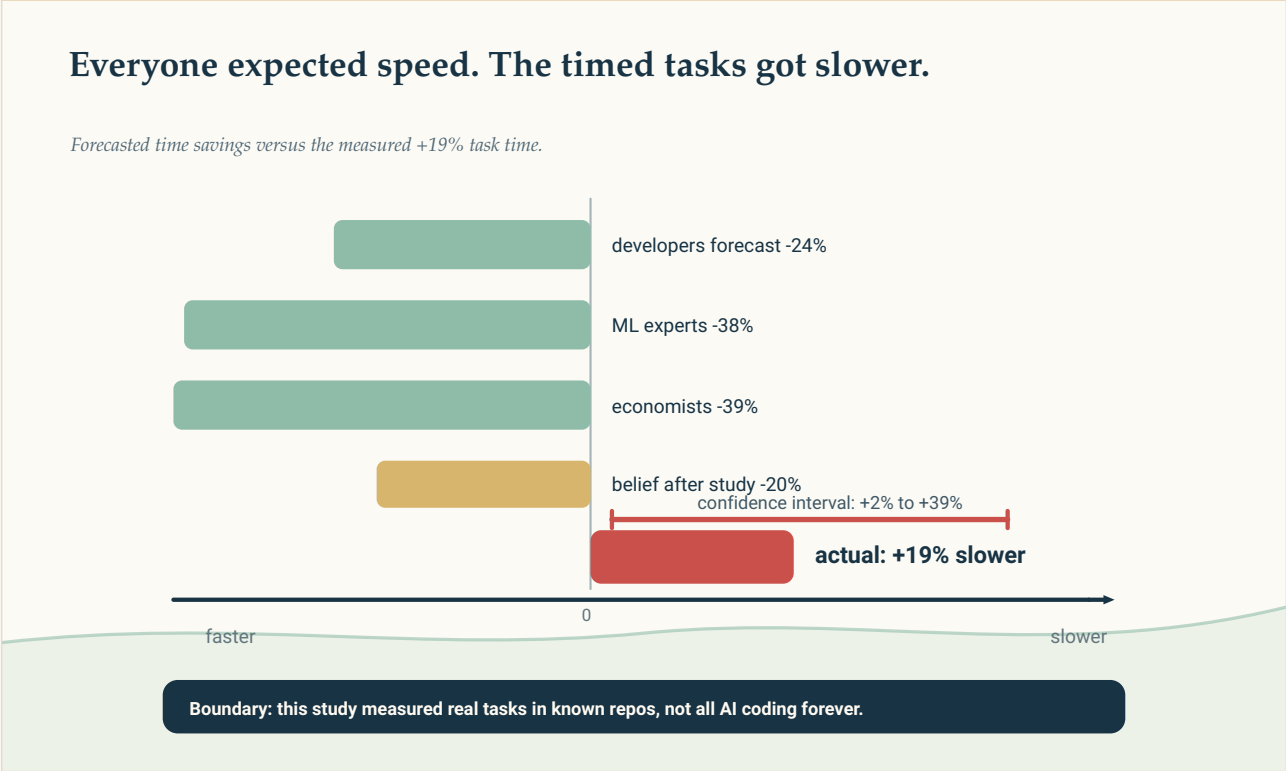
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Lập trình viên giàu kinh nghiệm cảm thấy nhanh hơn với AI trong khi đo được là chậm hơn — phát hiện thật, và không phải phán quyết về AI coding

Hỏi một lập trình viên giàu kinh nghiệm liệu trợ lý AI có làm họ nhanh hơn không, bạn thường sẽ nhận một con số: tiết kiệm cho tôi hai mươi, ba mươi phần trăm. Hỏi một nhà kinh tế hay nhà nghiên cứu machine learning, con số còn lớn hơn. Đầu năm 2025, một nhóm tại METR làm việc chậm và tốn kém hơn: kiểm tra thật. Họ lấy 16 lập trình viên open-source dày dạn, giao 246 tác vụ thật từ các codebase lớn mà họ biết rõ, rồi — tung đồng xu cho từng tác vụ — hoặc cho dùng công cụ AI, hoặc không. Sau đó họ bấm giờ.

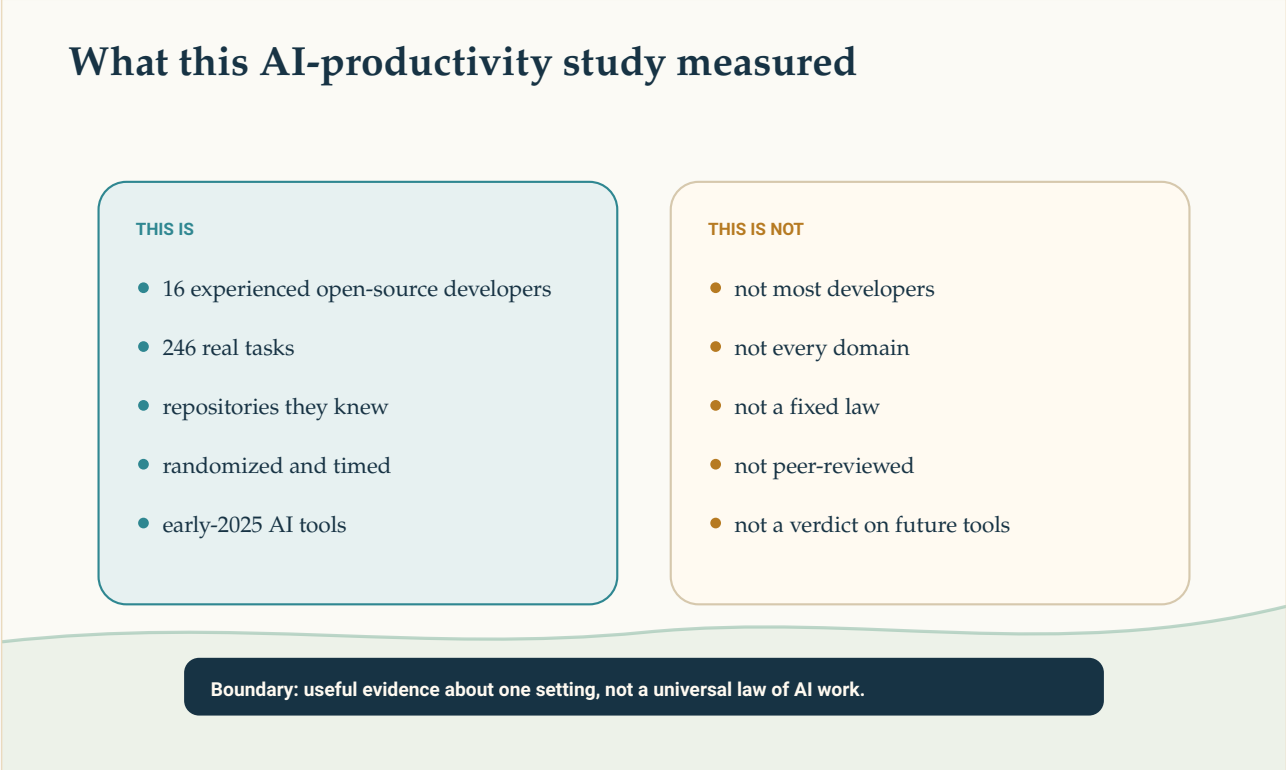
Các lập trình viên dự đoán AI sẽ giảm thời gian hoàn thành khoảng 24%. Kết quả ngược lại: các tác vụ có AI mất **lâu hơn 19%**. Và đây là phần đáng dừng lại — sau khi làm xong, chính những lập trình viên ấy vẫn tin AI đã làm họ nhanh hơn khoảng 20%. Họ chậm hơn nhưng cảm thấy nhanh hơn, và khoảng cách giữa hai con số đó là điều thú vị nhất của nghiên cứu.

Đây là một kết quả thật, được đo cẩn thận. Nhưng nó cũng chỉ là 16 lập trình viên, trên repository họ biết rất sâu, dùng công cụ đầu năm 2025 — và không thể nén thành câu "AI làm lập trình viên chậm hơn". Dữ liệu mới hơn của chính nhóm đã bắt đầu chỉ theo hướng ngược lại.



Mọi người đều dự đoán AI sẽ làm công việc nhanh hơn — lập trình viên −24%, chuyên gia ML −38%, nhà kinh tế −39%, và niềm tin sau thí nghiệm của chính lập trình viên −20%. Đồng hồ đo thấy +19% chậm hơn, với khoảng +2% đến +39%. Khoảng cách giữa cảm nhận và phép đo mới là phát hiện.

Original diagram — The Clean Paper CC BY 4.0



Nghiên cứu đo gì — 16 lập trình viên open-source chuyên gia, 246 tác vụ thật, repository quen thuộc, công cụ đầu năm 2025, phân ngẫu nhiên — và không đo gì: không đại diện cho phần lớn lập trình viên, không mọi lĩnh vực, không một quy luật cố định, chưa qua bình duyệt.

Original diagram — The Clean Paper CC BY 4.0

Nghiên cứu này đo gì, và ngẫu nhiên hóa mua cho ta điều gì

Một **thử nghiệm đối chứng ngẫu nhiên (RCT)** là công cụ y học thường dùng để phân biệt hiệu ứng thật khỏi hiệu ứng người ta hy vọng. Ở đây, mỗi trong 246 tác vụ được phân ngẫu nhiên vào điều kiện cho phép AI hoặc không, nên trung bình khác biệt có hệ thống duy nhất giữa hai nhóm là chính AI. Nhờ vậy có thể nói AI *gây ra* thay đổi về thời gian, thay vì chỉ nhận thấy những người chọn dùng AI tình cờ nhanh hay chậm vì lý do khác. Điều này quan trọng vì bằng chứng thông thường cho tăng năng suất coding bằng AI — tự báo cáo và điểm benchmark — không làm được: benchmark không phải công việc thật, và tự báo cáo, như nghiên cứu này cho thấy, có thể sai đầy tự tin. Các lập trình viên ở đây không phải người mới đang loay hoay với đồ chơi mới. Họ là contributor lâu năm cho các dự án open-source lớn, trưởng thành mà họ biết rõ, và đã có kinh nghiệm nhất định với công cụ.

Các tác giả đã làm gì

- Chạy một **RCT** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Mười sáu lập trình viên open-source giàu kinh nghiệm, mỗi người làm trên một repository lớn mà họ thường xuyên đóng góp và biết rõ — trung bình khoảng năm năm với dự án cụ thể đó.
- Dùng **246 tác vụ thật** — sửa bug, feature và refactor lấy từ issue tracker của chính các dự án. Mỗi tác vụ được phân ngẫu nhiên vào "AI allowed" hoặc "AI disallowed".
- "AI allowed" nghĩa là **tooling đầu năm 2025: Cursor Pro với Claude 3.5/3.7 Sonnet.** Thước đo chính là thời gian hoàn thành thật của mỗi tác vụ. Nhóm cũng thu dự đoán của lập trình viên trước khi làm, ước tính sau khi làm, cùng dự đoán từ chuyên gia kinh tế và machine learning.

Họ tìm thấy gì

- **Có AI, tác vụ mất lâu hơn 19%.** Không nhanh hơn — chậm hơn. Khoảng tin cậy 95% chạy từ khoảng +2% đến +39%, nên hướng kết quả khá vững dù độ lớn chính xác chưa chốt.
- **Mọi người dự đoán điều ngược lại.** Lập trình viên dự đoán nhanh hơn 24%; chuyên gia machine learning khoảng 38%; nhà kinh tế khoảng 39%. Cả ba nhóm kỳ vọng AI tiết kiệm nhiều thời gian; đồng hồ cho thấy nó tốn thêm.
- **Khoảng cách nhận thức.** Sau khi làm việc chậm hơn, lập trình viên vẫn ước tính AI đã tăng tốc khoảng 20% — chênh gần 40 điểm giữa điều họ cảm thấy và điều đồng hồ ghi.
- **Các nguyên nhân ứng viên được cân nhắc chứ chưa chứng minh.** Tác giả nêu các yếu tố có thể giải thích chậm đi: những lập trình viên này biết codebase của mình rất sâu nên trợ lý có ít thứ để bổ sung; dự án trưởng thành có tiêu chuẩn chất lượng cao, thường ngầm hiểu; repository lớn chứa nhiều ngữ cảnh mô hình không có; và thời gian thật bị tiêu vào prompting rồi review và sửa output AI. Đây là hướng giải thích, không phải phán quyết.

Điều nghiên cứu này không cho thấy

- Nó **không** cho thấy AI không tăng tốc phần lớn lập trình viên. Chính tác giả nói thẳng: 16 chuyên gia trên code họ biết thuộc lòng không phải lập trình viên trung bình trên tác vụ trung bình.
- Nó **không** cho thấy AI vô dụng, hay làm chậm trong các bối cảnh khác — người mới vào codebase, greenfield, ngôn ngữ chưa quen hoặc lĩnh vực khác.
- Nó **không** đóng băng công cụ tại chỗ. Đây là Cursor và Claude 3.5/3.7 Sonnet đầu năm 2025; tác giả nói rõ công cụ tốt hơn hoặc cách dùng tốt hơn có thể đổi kết quả ngay cả trong đúng bối cảnh này.
- Đây là **preprint** (đăng tháng 7/2025, chưa qua bình duyệt), và tác giả lưu ý họ không thể loại hoàn toàn artifact thực nghiệm — dù kết quả giữ qua các kiểm tra độ bền.
- Nó **không** cho phép cách đọc an ủi rằng lập trình viên chắc hẳn được lợi ở chỗ khác — học

nhiều hơn, vui hơn, viết code tốt hơn. Một điều được đo ở đây, cảm giác nhanh hơn, chính là điều dữ liệu phủ định.

Bằng chứng mạnh đến đâu

- **Thiết kế hiếm khi thành thật như vậy.** Ngẫu nhiên hóa, tác vụ thật, repository thật, đo thời gian thật — một bước lớn so với tự báo cáo và bảng benchmark mà phần lớn tuyên bố về AI coding dựa vào. Mức chậm 19% vượt qua các kiểm tra độ bền của tác giả.
- **Phát hiện dễ mang sang nơi khác nhất là khoảng cách nhận thức.** Trực giác chuyên gia về mức tăng tốc của chính mình với AI sai khoảng 40 điểm, theo hướng lạc quan. Đây là cảnh báo với *mọi* tăng năng suất AI dựa trên tự báo cáo — kể cả con số trong nghiên cứu này.
- **Đây là ảnh chụp, không phải xu hướng.** Follow-up tháng 2/2026 của chính METR, với cùng kiểu lập trình viên nhưng công cụ mới hơn, nghiêng về tăng tốc — rất thô khoảng −18% cho người quay lại và −4% cho người mới — nhưng tác giả đánh dấu bằng chứng yếu, méo bởi ai sẵn sàng tham gia (ngày càng nhiều lập trình viên từ chối làm không AI, mức trả giảm và chọn tác vụ lệch). Cách đọc trung thực là bức tranh đang chuyển động, và ngay cả chuyển động ấy cũng được báo cáo thận trọng.

Vì sao nó quan trọng

Phần lớn tranh luận về AI và lập trình chạy bằng demo và cảm giác: video màn hình mượt, tuyên bố tự tin, rồi cái trợn mắt phản đối. Điều hiếm — và khiến nghiên cứu đáng đọc — là ai đó làm thí nghiệm buồn tẻ: ngẫu nhiên hóa, bấm giờ công việc thật, rồi hỏi mọi người cảm thấy sao. Câu trả lời khó chịu cho cả hai phe. Nó chọc thủng câu chuyện AI đồng loạt tăng turbo cho lập trình viên chuyên gia trên code khó, quen thuộc. Và cũng chọc thủng câu chuyện đối lập gọn gàng — “AI làm lập trình viên chậm hơn, đã chứng minh” — vì dữ liệu mới của chính nhóm đã nghiêng ngược. Bài học bền nhất cũng nhỏ và rất người: người làm việc cảm thấy nhanh hơn trong khi đo được là chậm hơn. “Cảm giác nhanh hơn” không phải bằng chứng rằng nó nhanh hơn. Hãy đo.

Tóm tắt gọn

Trong một RCT, METR cho 16 lập trình viên open-source giàu kinh nghiệm hoàn thành 246 tác vụ thật trên codebase họ biết rõ, với công cụ AI đầu năm 2025 (Cursor Pro + Claude 3.5/3.7 Sonnet) được cho phép ngẫu nhiên ở một nửa tác vụ. Họ kỳ vọng AI giảm thời gian khoảng 24%; thực tế nó làm thời gian hoàn thành tăng 19% — và sau đó họ vẫn tin AI đã tăng tốc mình khoảng 20%. Khoảng cách nhận thức đó là lõi sắc và vững của nghiên cứu. Nhưng đây là 16 lập

trình viên, một bối cảnh hẹp và một ảnh chụp cố định đầu năm 2025; tác giả nói rõ nó không chứng minh AI không giúp phần lớn lập trình viên, và follow-up 2026 của chính họ đã nghiêng về tăng tốc, với lưu ý riêng. Một phép đo cẩn thận đáng xem nghiêm túc — không phải phán quyết về AI coding.

Nguồn

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/