



The CLEAN PAPER

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

Dezvoltatorii experimentați s-au simțit mai rapizi cu IA în timp ce lucrau măsurabil mai lent — rezultatul real, și verdictul despre coding cu IA care nu este

Un trial randomizat controlat de METR a pus șaisprezece dezvoltatori open-source experimentați să facă 246 de task-uri reale pe codebase-uri pe care le cunoșteau bine, cu instrumente IA de la începutul lui 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) permise pe o jumătate aleatorie. Ei se așteptau ca IA să reducă timpul pe task cu aproximativ 24%; în schimb, l-a crescut cu 19% — iar după aceea încă au crezut că îi accelerase cu aproximativ 20%. Acel decalaj de percepție este nucleul ascuțit și robust al studiului. Dar sunt șaisprezece dezvoltatori, un setting îngust și o fotografie fixă de la începutul lui 2025: autorii sunt expliciti că nu arată că IA nu îi ajută pe majoritatea dezvoltatorilor, iar follow-up-ul lor din 2026 înclină deja spre o accelerare, cu propriile caveat-uri.

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

Dezvoltatorii experimentați s-au simțit mai rapizi cu IA, deși lucrau măsurabil mai lent — rezultatul real și verdictul pe care studiul nu îl dă despre programarea cu IA

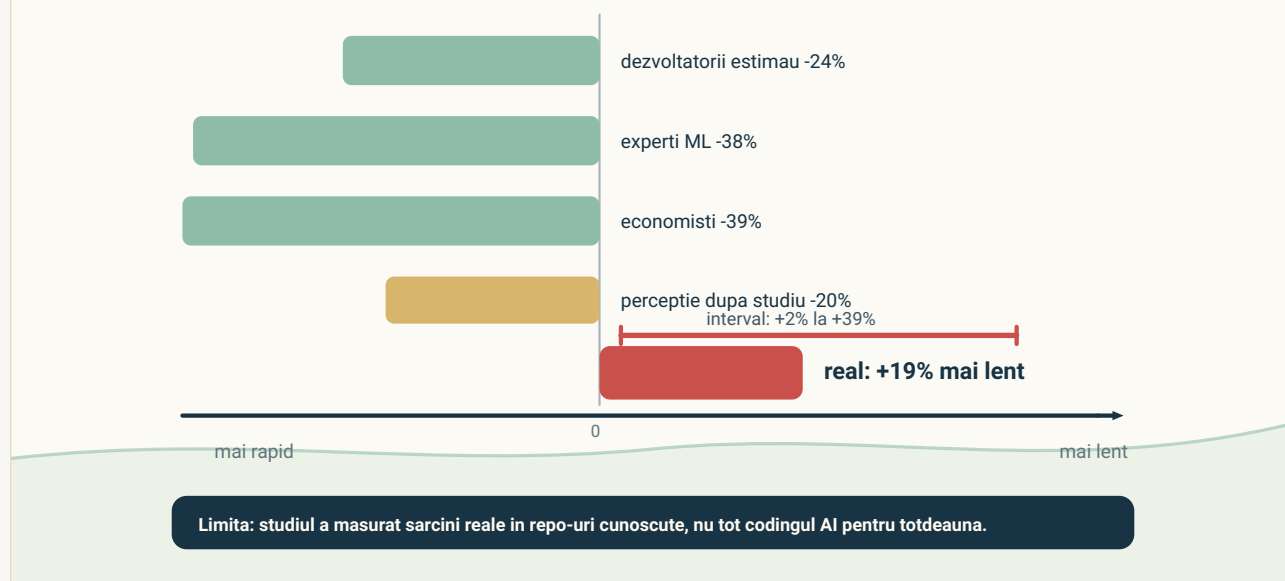
Întreabă un programator experimentat dacă un asistent de programare cu IA îl face mai rapid și vei primi adesea o estimare: îmi economisește douăzeci sau treizeci la sută din timp. Întreabă un economist ori un cercetător în învățare automată și estimarea poate fi și mai mare. La începutul lui 2025, o echipă de la METR a făcut partea lentă și costisitoare: a măsurat efectul. A recrutat șaisprezece dezvoltatori open-source cu experiență, le-a dat 246 de sarcini reale din baze de cod mari pe care le cunoșteau bine și a decis aleatoriu, pentru fiecare sarcină, dacă puteau sau nu să folosească instrumente IA. Apoi a cronometrat munca.

Dezvoltatorii estimaseră că IA le va reduce timpul de lucru cu aproximativ 24%. S-a întâmplat contrariul: sarcinile realizate cu IA au durat **cu 19% mai mult**. Partea cea mai interesantă vine după aceea: la final, aceiași dezvoltatori încă estimau că IA îi făcuse cu aproximativ 20% mai rapizi. Lucrau mai lent, dar aveau impresia că lucrează mai repede, iar diferența dintre percepție și măsurătoare este rezultatul cel mai remarcabil al studiului.

Este un rezultat real, măsurat cu grijă. Dar se referă la șaisprezece dezvoltatori care lucrau în depozite de cod pe care le cunoșteau foarte bine, folosind instrumentele disponibile la începutul lui 2025. Nu justifică afirmația generală „IA îi face pe dezvoltatori mai lenți”. Datele ulterioare ale aceleiași echipe indică deja o posibilă schimbare în sens opus.

Toti așteptau viteza. Sarcinile cronometrate au devenit mai lente.

Economii de timp prognozate versus +19% măsurat în timpul sarcinilor.



Toată lumea a prezis că IA va accelera munca — dezvoltatorii —24%, experții ML —38%, economiștii —39%, iar

convingerea de după lucru a dezvoltatorilor înșiși –20%. Cronometrul a găsit +19% mai lent, cu un interval de la +2% la +39%. Decalajul dintre perceput și măsurat este rezultatul.

Original diagram — The Clean Paper CC BY 4.0

Ce a masurat acest studiu despre productivitatea AI

ACESTA ESTE

- 16 dezvoltatori open-source experimentati
- 246 sarcini reale
- repo-uri pe care le cunosteau
- randomizat si cronometrat
- unelte AI inceput de 2025

ACESTA NU ESTE

- nu majoritatea dezvoltatorilor
- nu orice domeniu
- nu o lege fixa
- nu peer-reviewed
- nu un verdict despre unelte viitoare

Limita: dovada utila despre un context, nu o lege universala a muncii cu AI.

Ce măsoară studiul — 16 dezvoltatori open-source experimentați, 246 de sarcini reale, depozite de cod familiare, instrumente de la începutul lui 2025 și alocare aleatorie — și ce nu măsoară: majoritatea dezvoltatorilor, alte domenii sau o lege generală. Studiul era încă un preprint, neevaluat inter pares.

Original diagram — The Clean Paper CC BY 4.0

Ce a măsurat studiul și ce oferă un experiment randomizat

Un **experiment randomizat controlat (RCT)** este o metodă folosită pentru a deosebi un efect real de unul doar presupus. Aici, fiecare dintre cele 246 de sarcini a fost alocată aleatoriu unei condiții în care IA era permisă sau una în care nu era, astfel încât, în medie, singura diferență sistematică dintre cele două grupuri să fie utilizarea IA. Asta permite atribuirea diferenței de timp *utilizării IA*, în locul simplei observații că oamenii care aleg IA sunt mai rapizi sau mai lenți din alte motive. Este important deoarece dovezile obișnuite despre câștigurile de productivitate din programarea cu IA — autoevaluări și scoruri pe teste de referință — nu pot stabili acest lucru. Un test de referință nu este muncă reală, iar o autoevaluare poate fi sinceră și totuși greșită, după cum arată acest studiu. Participanții nu erau începători care încercau pentru prima dată o unealtă nouă, ci contributori constanți la proiecte open-source mari și mature, pe care le cunoșteau bine și pentru care aveau deja o anumită experiență cu instrumente IA.

Ce au făcut autorii

- Au desfășurat un **experiment randomizat controlat** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Șaisprezece dezvoltatori open-source experimentați, fiecare lucrând într-un depozit de cod mare la care contribuie regulat și pe care îl cunoaște bine — în medie de cinci ani pe proiectele respective.
- Au folosit **246 de sarcini reale** — corecții de erori, funcționalități noi și refactorizări preluate din sistemele de urmărire a problemelor ale propriilor proiecte. Fiecare sarcină a fost alocată aleatoriu la “IA permisă” sau “IA interzisă”.
- “IA permisă” însemna **instrumente disponibile la începutul lui 2025: Cursor Pro cu Claude 3.5/3.7 Sonnet**. Măsura principală a fost timpul real necesar pentru finalizarea fiecărei sarcini. În paralel, echipa a colectat predicții (de la dezvoltatori înainte) și estimări (de la ei după), plus predicții de la experți în economie și învățare automată.

Ce au găsit

- **Cu IA, sarcinile au durat cu 19% mai mult.** Nu mai repede — mai lent. Intervalul de încredere de 95% merge de la aproximativ +2% la +39%, deci direcția este solidă chiar dacă mărimea exactă nu este.
- **Toată lumea prezisese opusul.** Dezvoltatorii estimaseră o accelerare de 24%; experții în învățare automată, aproximativ 38%; economiștii, aproximativ 39%. Toate cele trei grupuri se așteptau ca IA să economisească mult timp; cronometrul a găsit că a costat timp.
- **Decalajul de percepție.** După ce au făcut munca și au ieșit mai lenți, dezvoltatorii încă estimau că IA îi accelerase cu aproximativ 20% — un decalaj de aproximativ 40 de puncte între ce simțeau și ce înregistra ceasul.
- **Motive candidate, cântărite mai degrabă decât dovedite.** Autorii aliniază factori care ar putea explica încetinirea: acești dezvoltatori își cunosc foarte bine bazele de cod, deci un asistent are mai puțin de adăugat; proiectele mature poartă standarde de calitate înalte, adesea implicite; depozitele de cod sunt mari și bogate în context pe care un model nu îl are; și o parte din timpul real se consumă pentru formularea prompturilor, apoi pentru revizuirea și corectarea rezultatului produs de IA. Le prezintă ca piste, nu ca verdicte.

Ce nu arată

- **Nu arată** că IA nu reușește să accelereze majoritatea dezvoltatorilor. Autorii o spun direct: șaisprezece experți pe cod pe care îl știu pe dinafară nu sunt dezvoltatorul mediu care rezolvă o sarcină obișnuită.
- **Nu arată** că IA este inutilă, sau că îi încetinește pe oameni în alte contexte — nou-veniți într-o bază de cod, proiecte începute de la zero (*greenfield*), limbaje nefamiliare sau domenii cu totul diferite.
- **Nu îngheață instrumentele în loc.** Acesta este Cursor și Claude 3.5/3.7 Sonnet de la începutul lui

2025; autorii sunt expliciti că instrumente mai bune, sau moduri mai bune de a le folosi pe acestea, ar putea schimba rezultatul chiar și în același context.

- Este un **preprint** (postat în iulie 2025, încă neevaluat inter pares), iar autorii notează că nu pot exclude complet artefacte experimentale — deși rezultatul a rezistat în analizele lor.
- **Nu** autorizează lectura liniștitoare că dezvoltatorii trebuie să fi câștigat altceva — au învățat mai mult, s-au simțit mai bine, au scris cod mai bun. Singurul lucru măsurat aici, accelerarea simțită, este exact ceea ce datele contrazic.

Cât de solide sunt dovezile?

- **Metodologia este neobișnuit de solidă.** Alocare aleatorie, sarcini reale, depozite de cod reale și cronometrare efectivă — un pas important dincolo de autoevaluările și clasamentele pe teste de referință pe care se sprijină multe afirmații despre programarea cu IA. Încetinirea de 19% a supraviețuit verificărilor de robustețe ale autorilor.
- **Rezultatul cel mai portabil este decalajul de percepție.** Intuiția experților despre propria accelerare cu IA a fost greșită cu aproximativ 40 de puncte, în direcția optimistă. Este un avertisment despre *orice* câștig de productivitate cu IA raportat de utilizatori — inclusiv numerele acestui studiu.
- **Este o fotografie de moment, nu o tendință.** Studiul ulterior al METR din februarie 2026, cu același tip de dezvoltatori pe instrumente mai noi, indică o accelerare — foarte aproximativ —18% pentru dezvoltatorii care au revenit și —4% pentru cei noi — dar autorii îl consideră o dovadă slabă, distorsionată de cine a fost dispus să participe (dezvoltatorii refuzau tot mai mult să lucreze fără IA, rata de plată a scăzut, iar selecția sarcinilor s-a modificat). Interpretarea prudentă este că situația se schimbă, iar chiar și această schimbare trebuie privită cu precauție din cauza selecției participanților.

De ce contează

O mare parte din discuția despre IA și programare se bazează pe demonstrații și impresii: o înregistrare de ecran elegantă, o afirmație sigură, o ridicare de ochi în direcția opusă. Ce este rar, și ce face studiul acesta demn de citit, este că cineva a făcut experimentul plictisitor — randomizează, cronometrează muncă reală, apoi întreabă oamenii cum a mers. Răspunsul este inconfortabil pentru ambele tabere. Sparge povestea că IA turbo-accelerează uniform dezvoltatorii experți pe cod greu și familiar. Și sparge și opusul ordonat — “IA îi face pe dezvoltatori mai lenți, dovedit” — pentru că datele mai noi ale aceleiași echipe înclină deja în cealaltă direcție. Lecția cea mai durabilă este și cea mai mică și mai umană: oamenii care făceau munca se simțeau mai rapizi în timp ce erau măsurabil mai lenți. “Se simte mai rapid” nu este dovadă că este. Măsoară.

Pe scurt

Într-un experiment randomizat controlat, METR a pus șaisprezece dezvoltatori open-source experimentați să finalizeze 246 de sarcini reale în baze de cod pe care le cunoșteau bine, cu instrumente IA de la începutul lui 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) permise pe o jumătate aleatorie. Dezvoltatorii se așteptau ca IA să le reducă timpul pe sarcină cu aproximativ 24%; în schimb, l-a crescut cu 19% — iar după aceea încă au crezut că îi accelerase cu aproximativ 20%. Acel decalaj de percepție este nucleul ascuțit și robust al studiului. Dar sunt șaisprezece dezvoltatori, un context restrâns și o fotografie fixă de la începutul lui 2025; autorii sunt expliți că nu arată că IA nu îi ajută pe majoritatea dezvoltatorilor, iar studiul ulterior al aceleiași echipe din 2026 indică deja o accelerare, cu propriile rezerve. O măsurare atentă care merită luată în serios — nu un verdict despre programarea cu IA.

Surse

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/>