



The CLEAN PAPER

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

Sviluppatori esperti si sono sentiti più veloci con l'IA mentre lavoravano misurabilmente più lentamente — il risultato reale, e non il verdetto sul coding con IA

Un trial randomizzato controllato di METR ha fatto svolgere a sedici sviluppatori open-source esperti 246 task reali su codebase che conoscevano bene, con strumenti IA dell'inizio 2025 (Cursor Pro più Claude 3.5/3.7 Sonnet) permessi su una metà casuale. Si aspettavano che l'IA riducesse il tempo dei task di circa il 24%; invece lo ha aumentato del 19% — e dopo credevano ancora che li avesse accelerati di circa il 20%. Quel divario di percezione è il nucleo netto e robusto dello studio. Ma sono sedici sviluppatori, un setting stretto e una fotografia fissa dell'inizio 2025: gli autori sono espliciti che non mostra che l'IA non aiuti la maggior parte degli sviluppatori, e il loro follow-up del 2026 punta già verso un'accelerazione, con caveat propri.

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

Sviluppatori esperti si sono sentiti più veloci con l'IA mentre lavoravano misurabilmente più lentamente — il risultato reale, e non il verdetto sulla programmazione con IA

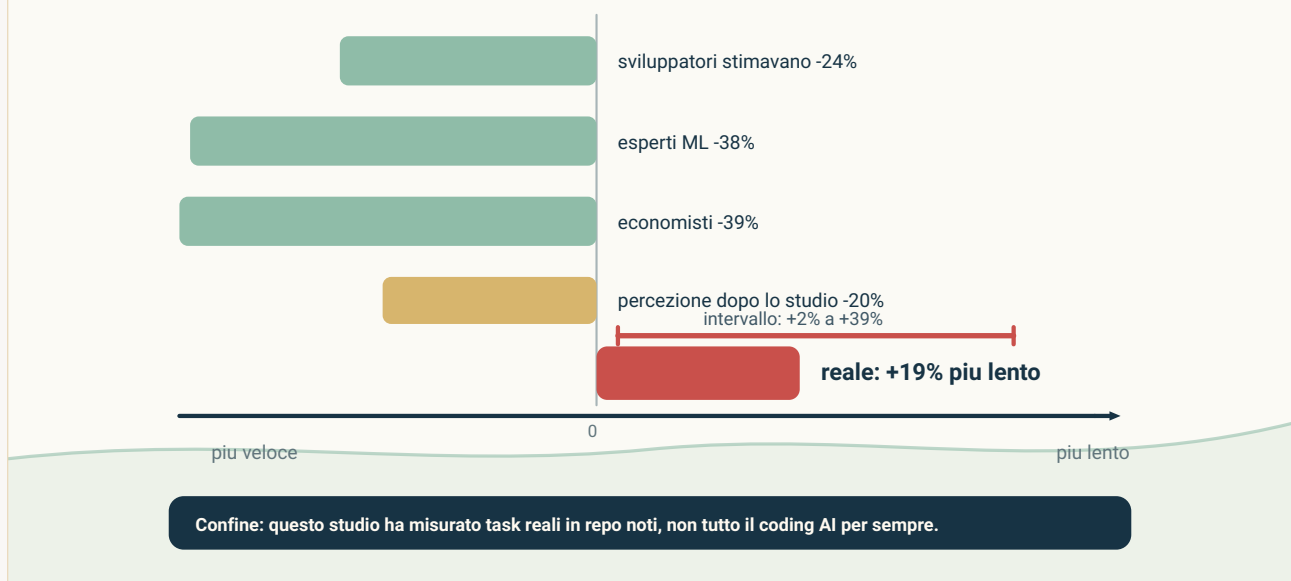
Chiedi a una programmatrice esperta se un assistente di coding con IA la renda più veloce e di solito otterrai un numero: mi fa risparmiare il venti, il trenta per cento. Chiedilo a un economista, o a un ricercatore di machine learning, e il numero cresce. All'inizio del 2025 un team di METR ha fatto la cosa lenta e costosa: ha controllato. Ha preso sedici sviluppatori open-source esperti, ha dato loro 246 task reali da grandi codebase che conoscevano bene e — lanciando una moneta, task per task — o ha permesso loro di usare strumenti IA o non l'ha permesso. Poi ha misurato il tempo.

Gli sviluppatori avevano previsto che l'IA avrebbe ridotto il tempo dei task di circa il 24%. Ha fatto l'opposto: i task svolti con l'IA hanno richiesto **il 19% di tempo in più**. E qui c'è la parte su cui vale la pena fermarsi — dopo aver finito, gli stessi sviluppatori credevano ancora che l'IA li avesse accelerati, di circa il 20%. Erano più lenti, e si sentivano più veloci, e la distanza tra quei due numeri è la cosa più interessante dello studio.

È un risultato reale, misurato con attenzione. È anche sedici sviluppatori, su repository che conoscono intimamente, con strumenti dell'inizio 2025 — e non è la frase piatta "l'IA rende gli sviluppatori più lenti". I dati successivi dello stesso team puntano già nell'altra direzione.

Tutti si aspettavano velocità. I task cronometrati sono diventati più lenti.

Risparmi di tempo previsti contro il +19% misurato nel tempo dei task.



Tutti avevano previsto che l'IA avrebbe accelerato il lavoro — sviluppatori -24%, esperti di ML -38%, economisti -39%, e la convinzione a posteriori degli stessi sviluppatori -20%. Il cronometro ha trovato +19% più lento, con un intervallo da +2% a +39%. Il divario tra percepito e misurato è il risultato.

Original diagram — The Clean Paper CC BY 4.0

Che cosa ha misurato questo studio sulla produttività AI

QUESTO E

- 16 sviluppatori open-source esperti
- 246 task reali
- repository che conoscevano
- randomizzato e cronometrato
- strumenti AI inizio 2025

QUESTO NON E

- non la maggior parte degli sviluppatori
- non ogni dominio
- non una legge fissa
- non peer-reviewed
- non un verdetto sugli strumenti futuri

Confine: evidenza utile su un setting, non una legge universale del lavoro con AI.

Che cosa misura lo studio — 16 sviluppatori open-source esperti, 246 task reali, repository familiari, strumenti dell'inizio 2025, randomizzazione — e che cosa non misura: non la maggior parte degli sviluppatori, non altri domini, non una legge fissa, non ancora sottoposto a revisione tra pari.

Original diagram — The Clean Paper CC BY 4.0

Che cosa ha misurato, e che cosa permette di stabilire uno studio randomizzato

Uno **studio randomizzato controllato (RCT)** è lo strumento che la medicina usa per distinguere un effetto reale da uno sperato. Qui, ciascuno dei 246 task è stato assegnato casualmente a essere svolto con IA permessa oppure no, così che in media l'unica differenza sistematica tra i due mucchi sia l'IA stessa. È questo che ti permette di dire che l'IA ha *causato* il cambiamento nel tempo, invece di limitarti a notare che le persone che si rivolgono all'IA sono più veloci o più lente per altri motivi. Conta perché le prove abituali sui guadagni del coding con IA — autovalutazioni e benchmark — non possono farlo: un benchmark non è lavoro reale, e un'autovalutazione, come mostra questo studio, può essere convintamente sbagliato. Gli sviluppatori qui non erano principianti che armeggiavano con un giocattolo nuovo. Erano contributor consolidati di grandi progetti open-source maturi, che conoscevano bene, con una certa esperienza precedente nell'uso degli strumenti.

Che cosa hanno fatto gli autori

- Hanno condotto uno **studio randomizzato controllato** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Sedici sviluppatori open-source esperti, ciascuno al lavoro su un grande repository a cui contribuisce regolarmente e che conosce bene — in media da cinque anni su quei progetti specifici.
- Hanno usato **246 task reali** — bug fix, feature e refactor presi dagli issue tracker degli stessi progetti. Ogni task è stato assegnato casualmente a “IA permessa” o “IA non permessa”.
- “IA permessa” significava **tooling dell’inizio 2025: Cursor Pro con Claude 3.5/3.7 Sonnet**. La misura principale era il tempo effettivo di completamento per task. In parallelo, il team ha raccolto previsioni (dagli sviluppatori prima) e stime (dagli stessi dopo), più previsioni da esperti di economia e machine learning.

Che cosa hanno trovato

- **Con l’IA, i task hanno richiesto il 19% di tempo in più.** Non più veloci — più lenti. L’intervallo di confidenza al 95% va da circa +2% a +39%, quindi la direzione è solida anche se la dimensione esatta non lo è.
- **Tutti avevano previsto il contrario.** Gli sviluppatori prevedevano un’accelerazione del 24%; gli esperti di machine learning circa il 38%; gli economisti circa il 39%. Tutti e tre i gruppi si aspettavano che l’IA facesse risparmiare molto tempo; il cronometro ha trovato che ne costava un po’.
- **Il divario di percezione.** Dopo aver fatto il lavoro ed essere risultati più lenti, gli sviluppatori stimavano ancora che l’IA li avesse accelerati di circa il 20% — un divario di circa 40 punti tra ciò che sentivano e ciò che il cronometro registrava.
- **Cause candidate, pesate ma non provate.** Gli autori mettono in fila fattori che potrebbero spiegare il rallentamento: questi sviluppatori conoscono profondamente le proprie codebase, quindi c’è meno che un assistente possa aggiungere; i progetti maturi portano standard di qualità alti e spesso impliciti; i repository sono grandi e pieni di contesto che un modello non ha; e tempo reale va nel prompting, poi nella revisione e correzione dell’output IA. Le presentano come piste, non come verdetti.

Che cosa non mostra

- **Non** mostra che l’IA non acceleri la maggior parte degli sviluppatori. Gli autori lo dicono direttamente: sedici esperti su codice che conoscono a memoria non sono lo sviluppatore medio sul task medio.
- **Non** mostra che l’IA sia inutile, o che rallenti le persone in altri contesti — nuovi arrivati in una codebase, lavoro greenfield, linguaggi non familiari, o campi del tutto diversi.
- **Non** congela gli strumenti nel tempo. Questo è Cursor e Claude 3.5/3.7 Sonnet all’inizio del 2025; gli autori sono espliciti sul fatto che strumenti migliori, o modi migliori di usare questi, potrebbero

cambiare il risultato anche nello stesso identico contesto.

- È un **preprint** (pubblicato a luglio 2025, non ancora sottoposto a revisione paritaria), e gli autori notano che non possono escludere del tutto artefatti sperimentali — anche se il risultato ha retto nelle loro analisi.
- **Non** autorizza la lettura rassicurante secondo cui gli sviluppatori devono aver guadagnato qualcos'altro — imparato di più, provato più soddisfazione, scritto codice migliore. L'unica cosa misurata qui, la sensazione di accelerazione, è esattamente ciò che i dati contraddicono.

Quanto sono solide le prove?

- **Il disegno è insolitamente onesto.** Randomizzato, task reali, repository reali, tempi effettivi — un grande passo avanti rispetto alle autovalutazioni e alle classifiche basate su benchmark su cui si reggono molte affermazioni sul coding con IA. Il rallentamento del 19% è sopravvissuto ai controlli di robustezza degli autori.
- **Il risultato più trasferibile è il divario di percezione.** L'intuizione di esperti sulla propria accelerazione dovuta all'IA era sbagliata di circa 40 punti, nella direzione ottimistica. È un avvertimento su *ogni* guadagno di produttività da IA dichiarato da chi lo usa — compresi i numeri di questo studio.
- **È una fotografia, non una tendenza.** Lo studio successivo di METR del febbraio 2026, con lo stesso tipo di sviluppatori su strumenti più nuovi, punta verso un'accelerazione — molto grossolanamente —18% per gli sviluppatori di ritorno e -4% per quelli nuovi — ma gli autori lo segnalano come un'indicazione debole, distorta da chi era disposto a partecipare (gli sviluppatori rifiutavano sempre più spesso di lavorare senza IA, la paga è scesa, e la selezione dei task si è spostata). La lettura onesta è che il quadro si sta muovendo, e anche il movimento viene riportato tenendo il pollice lontano dalla bilancia.

Perché conta

Gran parte della discussione su IA e programmazione vive di demo e sensazioni: una registrazione dello schermo ben confezionata, un'affermazione categorica, un'alzata di occhi opposta. Ciò che è raro, e che rende questo lavoro degno di lettura, è che qualcuno ha fatto l'esperimento noioso — randomizzare, misurare il tempo su lavoro reale, poi chiedere alle persone com'era andata. La risposta è scomoda per entrambi i campi. Fora la storia secondo cui l'IA turbo-carica uniformemente gli sviluppatori esperti su codice difficile e familiare. E fora anche l'opposto ordinato — “l'IA rende gli sviluppatori più lenti, dimostrato” — perché i dati più nuovi dello stesso team già pendono dall'altra parte. La lezione più duratura è anche la più piccola e la più umana: le persone che facevano il lavoro si sentivano più veloci mentre erano misurabilmente più lente. “Sembra più veloce” non è una prova che lo sia. Misuralo.

In breve

In uno studio randomizzato controllato, METR ha fatto completare a sedici sviluppatori open-source esperti 246 task reali su codebase che conoscevano bene, con strumenti IA dell'inizio 2025 (Cursor Pro più Claude 3.5/3.7 Sonnet) permessi su una metà casuale. Gli sviluppatori si aspettavano che l'IA riducesse il tempo dei task di circa il 24%; invece lo ha aumentato del 19% — e dopo credevano ancora che li avesse accelerati di circa il 20%. Quel divario di percezione è il nucleo netto e robusto dello studio. Ma sono sedici sviluppatori, un contesto molto specifico e una fotografia fissa dell'inizio 2025; gli autori sono espliciti che non mostra che l'IA non aiuti la maggior parte degli sviluppatori, e il loro studio successivo del 2026 punta già verso un'accelerazione, con le proprie cautele. Una misura attenta da prendere sul serio — non un verdetto sulla programmazione con IA.

Fonti

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/>