



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

Erfarne udviklere følte sig hurtigere med AI, mens de arbejdede målbart langsommere — det egentlige resultat, og hvorfor det ikke er en dom over AI-kodning

I et randomiseret kontrolleret forsøg lod METR seksten erfarne open source-udviklere udføre 246 virkelige opgaver i kodebaser, de kendte godt, med AI-værktøjer fra begyndelsen af 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) tilladt for en tilfældigt valgt halvdel. De forventede, at AI ville reducere tiden pr. opgave med omkring 24%; i stedet øgede den tiden med 19% — og bagefter troede de stadig, at AI havde gjort dem omkring 20% hurtigere. Denne opfattelseskluft er undersøgelsens skarpe og robuste kerne. Men det gælder seksten udviklere, én snæver situation og et fast øjebliksbillede fra begyndelsen af 2025: Forfatterne er tydelige om, at det ikke viser, at AI ikke hjælper de fleste udviklere, og deres egen opfølgning fra 2026 hælder allerede mod en tidsgevinst med sine egne forbehold.

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

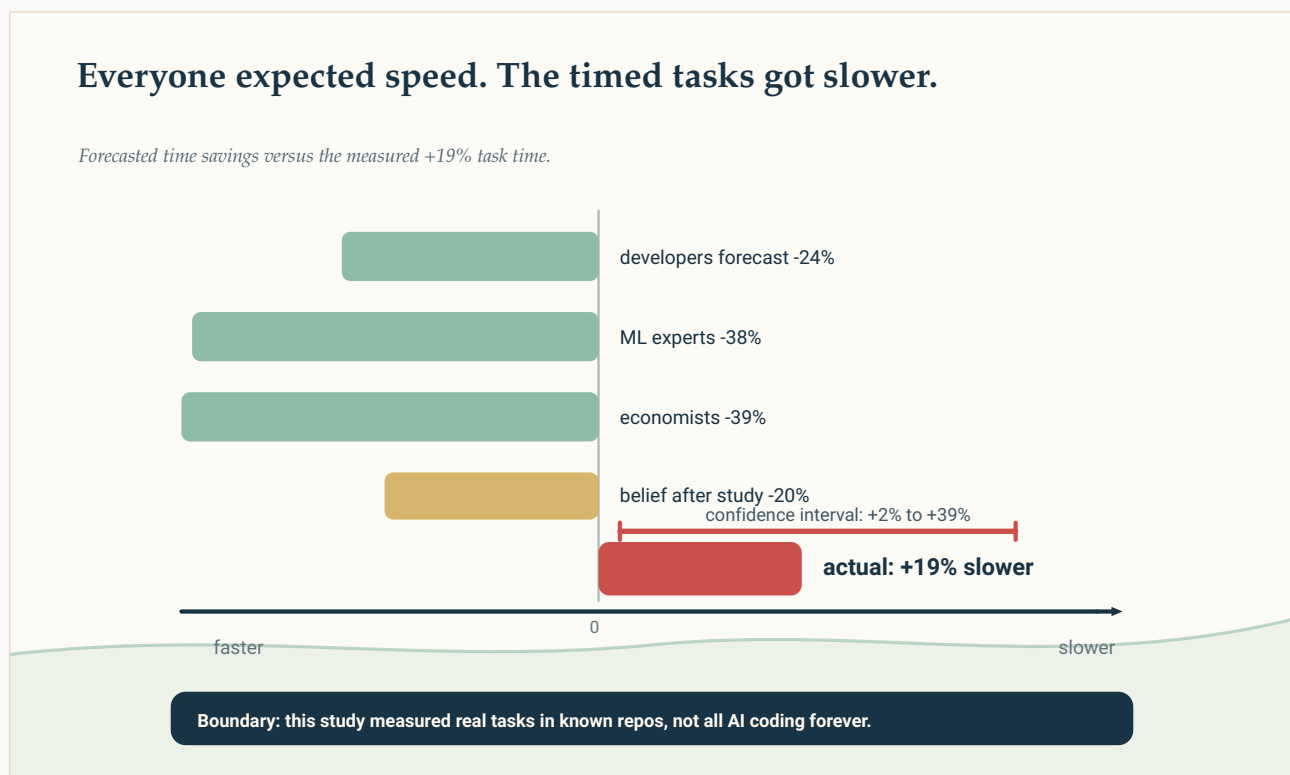
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Erfarne udviklere følte sig hurtigere med AI, mens de arbejdede målbart langsommere — det egentlige resultat, og hvorfor det ikke er en dom over AI-kodning

Spørg en erfaren programmør, om en AI-kodeassistent får arbejdet til at gå hurtigere, og du vil som regel få et tal til svar: Den sparer mig tyve, tredive procent. Spørg en økonom eller en maskinlæringsforsker, og tallet bliver større. I begyndelsen af 2025 gjorde et team hos METR det langsomme og dyre arbejde og undersøgte sagen. De tog seksten rutinerede open source-udviklere, gav dem 246 virkelige opgaver fra store kodebaser, de kendte godt, og lod dem — ved møntkast, opgave for opgave — enten bruge AI-værktøjer eller lade være. Derefter tog de tid på arbejdet.

Udviklerne havde forudsagt, at AI ville reducere tiden pr. opgave med omkring 24%. Det modsatte skete: Opgaverne udført med AI tog **19% længere tid**. Og her er den del, det er værd at dvæle ved — bagefter troede de samme udviklere stadig, at AI havde gjort dem omkring 20% hurtigere. De var langsommere og følte sig hurtigere, og afstanden mellem de to tal er det mest interessante ved undersøgelsen.

Det er et reelt og omhyggeligt målt resultat. Men det er også seksten udviklere, i kodearkiver, de kender ud og ind, med værktøjerne fra begyndelsen af 2025 — og det er ikke det kategoriske udsagn "AI gør udviklere langsommere". Senere data fra det samme team peger allerede i den anden retning.



Alle forudsagde, at AI ville få arbejdet til at gå hurtigere — udviklere –24%, ML-eksperter –38%, økonomer –39% og udviklernes egen opfattelse bagefter –20%. Stopuret viste +19% langsommere arbejde med et interval fra +2% til +39%. Forskellen mellem det oplevede og det målte er selve resultatet.

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.

Hvad undersøgelsen måler — 16 erfarne open source-udviklere, 246 virkelige opgaver, velkendte kodearkiver, værktøjer fra begyndelsen af 2025, randomiseret — og hvad den ikke viser: Resultaterne gælder ikke de fleste udviklere eller andre fagområder, er ikke en ufravigelig lov, og undersøgelsen er ikke fagfællebedømt.

Original diagram — The Clean Paper CC BY 4.0

Hvad dette målte, og hvad et randomiseret forsøg giver

Et **randomiseret kontrolleret forsøg (RCT)** er det redskab, lægevidenskaben bruger til at skelne en reel effekt fra en effekt, man blot håber på. Her blev hver af de 246 opgaver tilfældigt tildelt udførelse med AI tilladt eller ikke tilladt, så den eneste systematiske forskel mellem de to grupper i gennemsnit er AI'en selv. Det gør det muligt at sige, at AI *forårsagede* ændringen i tidsforbruget, frem for blot at bemærke, at mennesker, der vælger AI, tilfældigvis er hurtigere eller langsommere af andre grunde. Det er vigtigt, fordi de gængse belæg for produktivitetsgevinster ved AI-kodning — selvrapportering og benchmarkresultater — ikke kan gøre dette: En benchmark er ikke virkeligt arbejde, og en selvrapportering kan, som denne undersøgelse viser, være skråsikker og forkert. Udviklerne her var ikke nybegyndere, der fumlede med et nyt stykke legetøj. De var etablerede bidragydere til store, modne open source-projekter, som de kendte godt, og de havde en vis tidligere erfaring med at bruge værktøjerne.

Hvad forfatterne gjorde

- Gennemførte et **randomiseret kontrolleret forsøg** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Seksten erfarne open source-udviklere arbejdede hver især på et stort kodearkiv, som de regelmæssigt bidrager til og kender godt — i gennemsnit havde de arbejdet fem år på de konkrete projekter.
- Brugte **246 virkelige opgaver** — fejlrettelser, funktioner og refaktoreringer hentet fra projekternes egne opgavestyringssystemer. Hver opgave blev tilfældigt tildelt "AI tilladt" eller "AI ikke tilladt".
- "AI tilladt" betød **værktøjer fra begyndelsen af 2025: Cursor Pro med Claude 3.5/3.7 Sonnet**. Det primære mål var den faktiske tid, det tog at færdiggøre hver opgave. Derudover indsamlede teamet prognoser (fra udviklerne på forhånd) og skøn (fra dem bagefter) samt forudsigelser fra eksperter i økonomi og maskinlæring.

Hvad de fandt

- **Med AI tog opgaverne 19% længere tid.** Ikke kortere — længere. Konfidensintervallet på 95% går fra omtrent +2% til +39%, så retningen er sikker, selv om den præcise størrelse ikke er det.
- **Alle havde forudsagt det modsatte.** Udviklerne forudsagde en tidsgevinst på 24%, maskinlæringseksperter omkring 38% og økonomer omkring 39%. Alle tre grupper forventede, at AI ville spare en masse tid; stopuret viste, at det i stedet kostede tid.
- **Opfattelseskløften.** Efter at have udført arbejdet og være endt med at bruge længere tid vurderede udviklerne stadig, at AI havde gjort dem omkring 20% hurtigere — en forskel på omkring 40 procentpoint mellem det, de oplevede, og det, uret registrerede.
- **Mulige årsager, afvejte snarere end bevist.** Forfatterne fremhæver faktorer, der kan forklare nedgangen: Disse udviklere kender deres egne kodebaser indgående, så en assistent har mindre at bidrage med; modne projekter har høje kvalitetskrav, som ofte er underforståede; kodearkiverne er store og fulde af kontekst, en model ikke har; og der går reel tid med at formulere prompts og derefter gennemgå og rette AI-genererede resultater. De præsenterer dette som spor, ikke konklusioner.

Hvad dette ikke viser

- Det viser **ikke**, at AI ikke formår at gøre de fleste udviklere hurtigere. Forfatterne siger det direkte: Seksten eksperter, der arbejder med kode, de kan udenad, er ikke gennemsnitsudvikleren med gennemsnitsopgaven.
- Det viser **ikke**, at AI er ubrugelig, eller at den gør folk langsommere i andre situationer — nyttilkomne til en kodebase, arbejde fra bunden, ukendte sprog eller helt andre fagområder.
- Det **fastfryser ikke** værktøjerne. Dette er Cursor og Claude 3.5/3.7 Sonnet fra begyndelsen af 2025; forfatterne er tydelige om, at bedre værktøjer eller bedre måder at bruge disse på kan ændre

resultatet selv i præcis den samme situation.

- Det er et **preprint** (offentliggjort i juli 2025, endnu ikke fagfællebedømt), og forfatterne påpeger, at de ikke helt kan udelukke artefakter ved forsøget — selv om resultatet holdt i deres analyser.
- Det giver **ikke** belæg for den beroligende fortolkning, at udviklerne må have opnået en anden gevinst — lært mere, været gladere eller skrevet bedre kode. Den ene ting, der blev målt her, den oplevede tidsgevinst, er netop det, dataene modsiger.

Hvor stærke er belæggene

- **Forsøgsdesignet er usædvanligt ærligt.** Randomiseret, virkelige opgaver, virkelige kodearkiver, faktisk tidtagning — et stort skridt frem i forhold til den selvrapportering og de ranglister over benchmarkresultater, som de fleste påstande om AI-kodning bygger på. Tidsforøgelsen på 19% bestod forfatterens robusthedskontroller.
- **Det mest overførbare resultat er opfattelseskløften.** Eksperternes intuition om deres egen tidsgevinst med AI tog fejl med omkring 40 procentpoint i den optimistiske retning. Det er en advarsel om *enhver* selvrapporteret produktivitsgevinst fra AI — inklusive denne undersøgelses egne tal.
- **Det er et øjebliksbillede, ikke en trend.** METR's egen opfølgning fra februar 2026, med den samme type udviklere og nyere værktøjer, peger mod en tidsgevinst — meget groft –18% for tilbagevendende udviklere og –4% for nye — men forfatterne betegner det som svage belæg, fordrejet af, hvem der var villige til at deltage (udviklere afslog i stigende grad at arbejde uden AI, betalingssatsen faldt, og udvælgelsen af opgaver blev skæv). Den ærlige fortolkning er, at billedet bevæger sig, og at selv bevægelsen rapporteres uden en finger på vægtskålen.

Hvorfor det er vigtigt

Det meste af diskussionen om AI og programmering bygger på demonstrationer og maveførmelser: en elegant skærmoptagelse, en skråsikker påstand, et afvisende rul med øjnene. Det sjældne, og det, der gør dette værd at læse, er, at nogen gennemførte det kedelige eksperiment — randomiser, tag tid på virkeligt arbejde, og spørg så folk, hvordan det gik. Svaret er ubehageligt for begge lejre. Det punkterer fortællingen om, at AI altid sætter turbo på erfarne udviklere, når de arbejder med svær, velkendt kode. Og det punkterer den enkle modsætning — "AI gør udviklere langsommere, bevist" — fordi det samme teams nyere data allerede hælder i den anden retning. Den mest holdbare lære er også den mindste og mest menneskelige: De mennesker, der udførte arbejdet, følte sig hurtigere, mens de målbart var langsommere. "Det føles hurtigere" er ikke belæg for, at det er det. Mål det.

Kort fortalt

I et randomiseret kontrolleret forsøg lod METR seksten erfarne open source-udviklere færdiggøre 246 virkelige opgaver i kodebaser, de kendte godt, med AI-værktøjer fra begyndelsen af 2025 (Cursor

Pro plus Claude 3.5/3.7 Sonnet) tilladt for en tilfældigt valgt halvdel. Udviklerne forventede, at AI ville reducere tiden pr. opgave med omkring 24%; i stedet øgede den tiden med 19% — og bagefter troede de stadig, at AI havde gjort dem omkring 20% hurtigere. Denne opfattelseskløft er undersøgelsens skarpe og robuste kerne. Men det gælder seksten udviklere, én snæver situation og et fast øjebliksbillede fra begyndelsen af 2025; forfatterne er tydelige om, at det ikke viser, at AI ikke hjælper de fleste udviklere, og deres egen opfølgning fra 2026 peger allerede mod en tidsgevinst med sine egne forbehold. En omhyggelig måling, der er værd at tage alvorligt — ikke en dom over AI-kodning.

Kilder

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/>