



The CLEAN PAPER

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

Erfarne utviklere følte seg raskere med KI mens de jobbet målbart saktere — det egentlige funnet, og hvorfor det ikke er en dom over KI-koding

I en randomisert kontrollert studie lot METR seksten erfarne utviklere av åpen kildekode utføre 246 reelle oppgaver i kodebaser de kjente godt, med KI-verktøy fra tidlig i 2025 (Cursor Pro pluss Claude 3.5/3.7 Sonnet) tillatt for en tilfeldig valgt halvpart. De ventet at KI ville redusere tiden per oppgave med rundt 24%; i stedet økte den tiden med 19% — og etterpå trodde de fortsatt at KI hadde gjort dem rundt 20% raskere. Dette oppfatningsgapet er den skarpe, robuste kjernen i studien. Men det gjelder seksten utviklere, én snever situasjon og et fast øyeblikksbilde fra tidlig i 2025: Forfatterne er tydelige på at det ikke viser at KI ikke hjelper de fleste utviklere, og deres egen oppfølging fra 2026 heller allerede mot 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

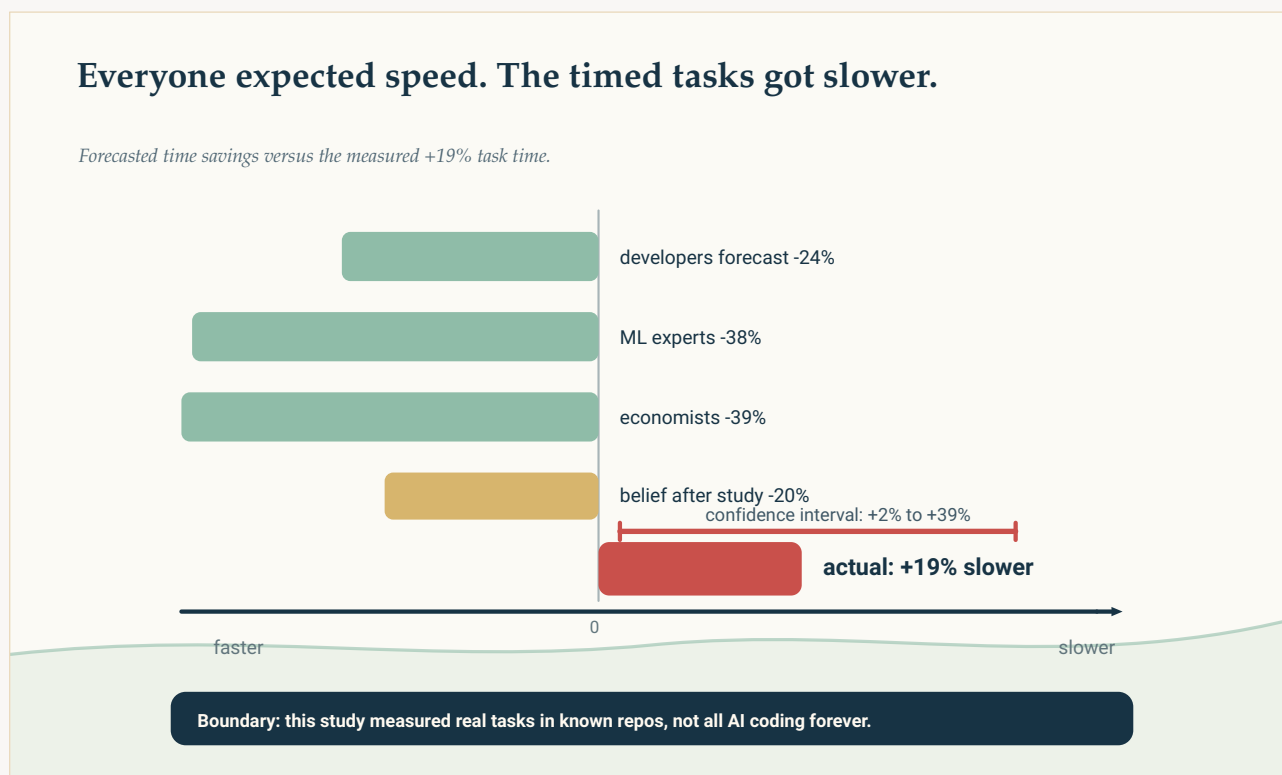
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Erfarne utviklere følte seg raskere med KI mens de jobbet målbart saktere — det egentlige funnet, og hvorfor det ikke er en dom over KI-koding

Spør en erfaren programmerer om en KI-kodeassistent gjør arbeidet raskere, og du får som regel et tall til svar: Den sparer meg tjue, tretti prosent. Spør en økonom eller en maskinlæringsforsker, og tallet blir høyere. Tidlig i 2025 gjorde et team ved METR det langsomme og kostbare arbeidet og undersøkte saken. De tok seksten rutinerte utviklere av åpen kildekode, ga dem 246 reelle oppgaver fra store kodebaser de kjente godt, og lot dem — ved myntkast, oppgave for oppgave — enten bruke KI-verktøy eller la være. Deretter tok de tiden på arbeidet.

Utviklerne hadde anslått at KI ville redusere tiden per oppgave med rundt 24%. Det motsatte skjedde: Oppgavene som ble gjort med KI, tok **19% lengre tid**. Og her er delen det er verdt å dvele ved — etterpå trodde de samme utviklerne fortsatt at KI hadde gjort dem rundt 20% raskere. De var langsommere og følte seg raskere, og avstanden mellom disse to tallene er det mest interessante ved studien.

Dette er et reelt, grundig målt resultat. Men det er også seksten utviklere, i kodearkiver de kjenner ut og inn, med verktøy fra tidlig i 2025 — og det er ikke det kategoriske utsagnet «KI gjør utviklere langsommere». Senere data fra det samme teamet peker allerede i motsatt retning.



Alle anslo at KI ville få arbeidet til å gå raskere — utviklere –24%, ML-eksperter –38%, økonomer –39% og utviklernes egen oppfatning i etterkant –20%. Stoppeklokken viste +19% langsommere arbeid, med et intervall fra +2% til +39%. Gapet mellom opplevd og målt er selve funnet.

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.

Hva studien måler — 16 erfarne utviklere av åpen kildekode, 246 reelle oppgaver, velkjente kodearkiver, verktøy fra tidlig i 2025, randomisert — og hva den ikke viser: Resultatene gjelder ikke de fleste utviklere eller andre fagområder, er ingen uforanderlig lov, og studien er ikke fagfellevurdert.

Original diagram — The Clean Paper CC BY 4.0

Hva dette målte, og hva en randomisert studie gir

En **randomisert kontrollert studie (RCT)** er verktøyet medisinen bruker for å skille en reell effekt fra en effekt man bare håper på. Her ble hver av de 246 oppgavene tilfeldig tildelt gjennomføring med KI tillatt eller ikke tillatt, slik at den eneste systematiske forskjellen mellom de to gruppene i gjennomsnitt er KI i seg selv. Det er dette som gjør at man kan si at KI *forårsaket* endringen i tidsbruk, i stedet for bare å legge merke til at personer som velger KI, tilfeldigvis er raskere eller langsommere av andre grunner. Det er viktig fordi de vanlige bevisene for produktivetsgevinst ved KI-koding — egenrapportering og resultater på standardiserte tester — ikke kan gjøre dette: En slik test er ikke reelt arbeid, og en egenrapportering kan, som denne studien viser, være skråsikker og feil. Utviklerne her var ikke nybegynnere som fomlet med et nytt leketøy. De var etablerte bidragsytere til store, modne prosjekter med åpen kildekode som de kjente godt, og de hadde noe tidligere erfaring med å bruke verktøyene.

Hva forfatterne gjorde

- Gjennomførte en **randomisert kontrollert studie** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Seksten erfarne utviklere av åpen kildekode jobbet hver for seg med et stort kodearkiv som de regelmessig bidrar til og kjenner godt — i gjennomsnitt hadde de jobbet fem år med de konkrete prosjektene.
- Brukte **246 reelle oppgaver** — feilrettinger, funksjoner og refaktoreringer hentet fra prosjektenes egne sakssystemer. Hver oppgave ble tilfeldig tildelt «KI tillatt» eller «KI ikke tillatt».
- «KI tillatt» innebar **verktøy fra tidlig i 2025: Cursor Pro med Claude 3.5/3.7 Sonnet**. Hovedmålet var den faktiske tiden det tok å fullføre hver oppgave. I tillegg samlet teamet inn prognoser (fra utviklerne på forhånd) og anslag (fra dem etterpå), samt forutsigelser fra eksperter på økonomi og maskinlæring.

Hva de fant

- **Med KI tok oppgavene 19% lengre tid.** Ikke kortere — lengre. Konfidensintervallet på 95% går fra omtrent +2% til +39%, så retningen er sikker selv om den nøyaktige størrelsen ikke er det.
- **Alle hadde forutsagt det motsatte.** Utviklerne anslo en tidsgevinst på 24%, maskinlæringseksperterene rundt 38% og økonomene rundt 39%. Alle tre gruppene ventet at KI ville spare mye tid; stoppeklokken viste at det i stedet kostet tid.
- **Oppfatningsgapet.** Etter å ha gjort arbeidet og endt opp med å bruke lengre tid, anslo utviklerne fortsatt at KI hadde gjort dem rundt 20% raskere — et gap på omtrent 40 prosentpoeng mellom det de følte, og det klokken registrerte.
- **Mulige årsaker, vurdert snarere enn bevist.** Forfatterne trekker frem faktorer som kan forklare nedgangen: Disse utviklerne kjenner sine egne kodebaser inngående, så en assistent har mindre å tilføre; modne prosjekter har høye kvalitetskrav som ofte er underforståtte; kodearkivene er store og fulle av kontekst en modell ikke har; og det går reell tid med til å skrive prompter og deretter gjennomgå og rette opp KI-genererte resultater. De presenterer dette som spor, ikke konklusjoner.

Hva dette ikke beviser

- Det viser **ikke** at KI ikke gjør de fleste utviklere raskere. Forfatterne sier det direkte: Seksten eksperter som arbeider med kode de kan utenat, er ikke gjennomsnittsutvikleren med gjennomsnittsopgaven.
- Det viser **ikke** at KI er ubrukelig, eller at den gjør folk langsommere i andre situasjoner — nykommere i en kodebase, arbeid fra bunnen av, ukjente språk eller helt andre fagområder.
- Det forutsetter **ikke** at verktøyene står stille. Dette er Cursor og Claude 3.5/3.7 Sonnet fra tidlig i 2025; forfatterne er tydelige på at bedre verktøy, eller bedre måter å bruke disse på, kan endre resultatet selv i nøyaktig den samme situasjonen.

- Det er et **preprint** (lagt ut i juli 2025, ennå ikke fagfellevurdert), og forfatterne påpeker at de ikke helt kan utelukke artefakter ved eksperimentet — selv om funnet sto seg i analysene deres.
- Det gir **ikke** grunnlag for den beroligende tolkningen at utviklerne må ha oppnådd en annen gevinst — lært mer, vært gladere eller skrevet bedre kode. Det ene som ble målt her, den opplevde tidsgevinsten, er nettopp det dataene motsier.

Hvor sterke er bevisene?

- **Forsøksopplegget er uvanlig ærlig.** Randomisert, reelle oppgaver, reelle kodearkiver, faktisk tidtaking — et stort skritt fremover fra egenrapporteringene og resultatlistene fra standardiserte tester som de fleste påstander om KI-koding bygger på. Tidsøkningen på 19% besto forfatterenes robusthetkontroller.
- **Det mest overførbare funnet er oppfatningsgapet.** Ekspertenenes intuisjon om sin egen tidsgevinst med KI tok feil med rundt 40 prosentpoeng, i optimistisk retning. Det er en advarsel om *enhver* egenrapportert produktivitetsgevinst fra KI — inkludert tallene fra denne studien.
- **Det er et øyeblikksbilde, ikke en trend.** METRs egen oppfølgingsstudie fra februar 2026, med den samme typen utviklere og nyere verktøy, peker mot en tidsgevinst — svært grovt –18% for tilbakevendende utviklere og –4% for nye — men forfatterne beskriver det som svake bevis, fordreid av hvem som var villige til å delta (utviklere avsto i økende grad å jobbe uten KI, betalingssatsen falt, og utvalget av oppgaver ble skjevt). Den ærlige tolkningen er at bildet er i bevegelse, og at selv bevegelsen rapporteres uten en tommel på vektskålen.

Hvorfor det er viktig

Det meste av diskusjonen om KI og programmering bygger på demonstrasjoner og magefølelse: et elegant skjermopptak, en skråsikker påstand, et avvisende himlende blikk. Det som er sjeldent, og som gjør dette verdt å lese, er at noen gjennomførte det kjedelige eksperimentet — randomiser, ta tiden på reelt arbeid, og spør deretter folk hvordan det gikk. Svaret er ubehagelig for begge leirer. Det punkterer fortellingen om at KI alltid gir erfarne utviklere turbofart på vanskelig, velkjent kode. Og det punkterer den ryddige motsetningen — «KI gjør utviklere langsommere, bevist» — fordi det samme teamets nyere data allerede heller i motsatt retning. Den mest varige lærdommen er også den minste og mest menneskelige: De som gjorde arbeidet, følte seg raskere mens de målbart var langsommere. «Det føles raskere» er ikke bevis på at det er det. Mål det.

Kort oppsummert

I en randomisert kontrollert studie lot METR seksten erfarne utviklere av åpen kildekode fullføre 246 reelle oppgaver i kodebaser de kjente godt, med KI-verktøy fra tidlig i 2025 (Cursor Pro pluss Claude 3.5/3.7 Sonnet) tillatt for en tilfeldig valgt halvpart. Utviklerne ventet at KI ville redusere tiden per

oppgave med rundt 24%; i stedet økte den tiden med 19% — og etterpå trodde de fortsatt at KI hadde gjort dem rundt 20% raskere. Dette oppfatningsgapet er den skarpe, robuste kjernen i studien. Men det gjelder seksten utviklere, én snever situasjon og et fast øyeblikksbilde fra tidlig i 2025; forfatterne er tydelige på at det ikke viser at KI ikke hjelper de fleste utviklere, og deres egen oppfølging fra 2026 peker allerede mot en tidsgevinst, med sine egne forbehold. En grundig måling det er verdt å ta på alvor — ikke en dom over KI-koding.

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/>