



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

Erfarna utvecklare kände sig snabbare med AI samtidigt som de arbetade mätbart långsammare — det verkliga resultatet, och varför det inte är en dom över AI-kodning

I en randomiserad kontrollerad studie lät METR sexton erfarna utvecklare inom öppen källkod utföra 246 verkliga uppgifter i kodbaser som de kände väl, med AI-verktyg från början av 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) tillåtna för en slumpmässigt vald hälft. De väntade sig att AI skulle minska tiden per uppgift med omkring 24%; i stället ökade den tiden med 19% — och efteråt trodde de fortfarande att AI hade gjort dem omkring 20% snabbare. Detta uppfattningsgap är studiens skarpa, robusta kärna. Men det gäller sexton utvecklare, en enda snäv situation och en fixerad ögonblicksbild från början av 2025: författarna är tydliga med att det inte visar att AI misslyckas med att hjälpa de flesta utvecklare, och deras egen uppföljning från 2026 lutar redan mot en tidsvinst, med egna förbehåll.

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

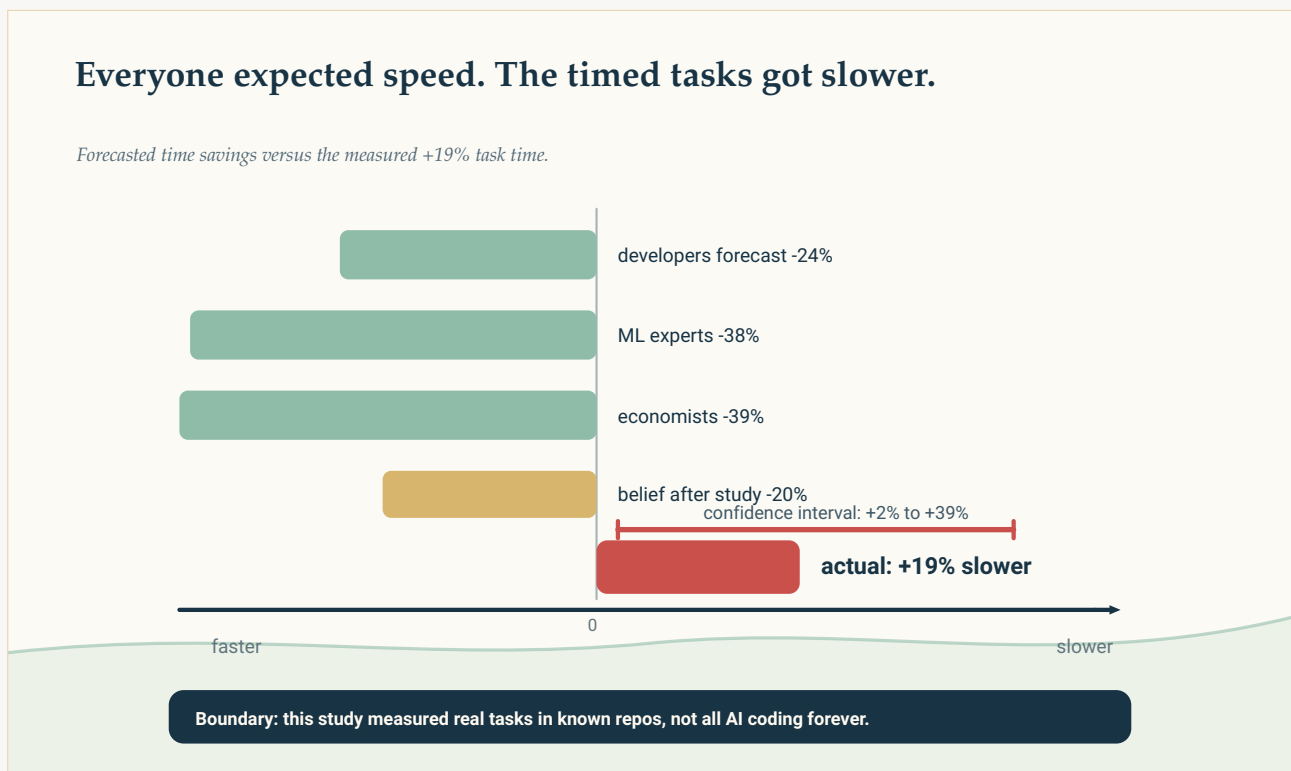
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Erfarna utvecklare kände sig snabbare med AI samtidigt som de arbetade mätbart långsammare — det verkliga resultatet, och varför det inte är en dom över AI-kodning

Fråga en erfaren programmerare om en AI-kodassistent gör arbetet snabbare, så får du oftast en siffra till svar: den sparar mig tjugo, trettio procent. Fråga en ekonom eller en maskininlärningsforskare, så blir siffran högre. I början av 2025 gjorde ett team på METR det långsamma och kostsamma arbetet och kontrollerade saken. De tog sexton rutinerade utvecklare inom öppen källkod, gav dem 246 verkliga uppgifter från stora kodbaserna som de kände väl och lät dem — genom slantsingling, uppgift för uppgift — antingen använda AI-verktyg eller avstå från dem. Sedan tog de tid på arbetet.

Utvecklarna hade förutspått att AI skulle minska tiden per uppgift med omkring 24%. Det blev tvärtom: uppgifterna som gjordes med AI tog **19% längre tid**. Och här är den del som är värd att dröja vid — efteråt trodde samma utvecklare fortfarande att AI hade gjort dem omkring 20% snabbare. De var långsammare och kände sig snabbare, och avståndet mellan dessa två siffror är det mest intressanta i studien.

Det här är ett verkligt, noggrant uppmätt resultat. Men det gäller också sexton utvecklare, i kodförråd som de känner utan och innan, med verktyg från början av 2025 — och det är inte det kategoriska påståendet "AI gör utvecklare långsammare". Senare data från samma team pekar redan åt andra hållet.



Alla förutspådde att AI skulle påskynda arbetet — utvecklare –24%, ML-experter –38%, ekonomer –39% och utvecklarnas egen uppfattning i efterhand –20%. Tidtagningen visade +19% långsammare arbete, med ett intervall på +2% till +39%. Skillnaden mellan upplevt och uppmätt är 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.

Vad studien mäter — 16 erfarna utvecklare inom öppen källkod, 246 verkliga uppgifter, välkända kodförråd, verktyg från början av 2025, randomiserat — och vad den inte visar: resultaten gäller inte de flesta utvecklare eller andra områden, är ingen oföränderlig lag och studien är inte sakkunniggranskad.

Original diagram — The Clean Paper CC BY 4.0

Vad detta mätte och vad en randomiserad studie ger

En **randomiserad kontrollerad studie (RCT)** är det verktyg som medicinen använder för att skilja en verklig effekt från en önskad sådan. Här lottades var och en av de 246 uppgifterna till att utföras med AI tillåten eller utan AI, så att den enda systematiska skillnaden mellan de två grupperna i genomsnitt är AI i sig. Det är detta som gör det möjligt att säga att AI *orsakade* förändringen i tidsåtgång, i stället för att bara konstatera att personer som väljer AI av andra skäl råkar vara snabbare eller långsammare. Det spelar roll eftersom de vanliga beläggen för produktivitetsvinster med AI-kodning — självrapporter och resultat på standardiserade tester — inte kan göra detta: ett sådant test är inte verkligt arbete, och en självrapport kan, som den här studien visar, vara tvärsäker och felaktig. Utvecklarna här var inte nybörjare som fumlade med en ny leksak. De var etablerade bidragsgivare till stora, mogna projekt med öppen källkod som de kände väl och hade viss tidigare erfarenhet av att använda verktygen.

Vad författarna gjorde

- Genomförde en **randomiserad kontrollerad studie** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Sexton erfarna utvecklare inom öppen källkod arbetade var och en i ett stort kodförråd som de regelbundet bidrar till och känner väl — i genomsnitt hade de arbetat fem år med de specifika projekten.
- Använde **246 verkliga uppgifter** — felrättningar, funktioner och omstruktureringar hämtade från projektens egna ärendehanteringssystem. Varje uppgift lottades till "AI tillåten" eller "AI inte tillåten".
- "AI tillåten" innebar **verktyg från början av 2025: Cursor Pro med Claude 3.5/3.7 Sonnet**. Det huvudsakliga måttet var den faktiska tiden för att slutföra varje uppgift. Dessutom samlade teamet in prognoser (från utvecklarna i förväg) och uppskattningar (från dem efteråt), samt förutsägelser från experter inom ekonomi och maskininlärning.

Vad de fann

- **Med AI tog uppgifterna 19% längre tid.** Inte kortare — längre. Konfidensintervallet på 95% sträcker sig från ungefär +2% till +39%, så riktningen är säker även om den exakta storleken inte är det.
- **Alla hade förutspått motsatsen.** Utvecklarna förutspådde en tidsvinst på 24%, maskininläringsexperter omkring 38% och ekonomer omkring 39%. Alla tre grupper väntade sig att AI skulle spara mycket tid; tidtagningen visade att det i stället kostade tid.
- **Uppfattningsgapet.** Efter att ha gjort arbetet och blivit långsammare uppskattade utvecklarna fortfarande att AI hade gjort dem omkring 20% snabbare — ett gap på ungefär 40 procentenheter mellan vad de upplevde och vad klockan registrerade.
- **Möjliga förklaringar, värderade snarare än bevisade.** Författarna radar upp faktorer som skulle kunna förklara nedgången: dessa utvecklare känner sina egna kodbaser på djupet, så en assistent har mindre att tillföra; mogna projekt har höga kvalitetskrav som ofta är underförstådda; kodförråden är stora och fulla av sammanhang som en modell saknar; och verklig tid går åt till att formulera promptar och sedan granska och korrigera AI-genererat resultat. De presenterar detta som spår, inte slutsatser.

Vad detta inte visar

- Det visar **inte** att AI misslyckas med att göra de flesta utvecklare snabbare. Författarna säger det direkt: sexton experter som arbetar med kod de kan utantill är inte den genomsnittliga utvecklaren med den genomsnittliga uppgiften.
- Det visar **inte** att AI är oanvändbar eller att den gör människor långsammare i andra situationer — för nykomlingar i en kodbas, arbete från grunden, obekanta språk eller helt andra områden.
- Det låser **inte** verktygen i tiden. Det här är Cursor och Claude 3.5/3.7 Sonnet från början av 2025;

författarna är tydliga med att bättre verktyg, eller bättre sätt att använda dessa, skulle kunna ändra resultatet även i exakt samma situation.

- Det är ett **preprint** (publicerat i juli 2025, ännu inte sakkunniggranskat), och författarna påpekar att de inte helt kan utesluta artefakter från experimentet — även om resultatet stod sig i deras analyser.
- Det ger **inte** stöd för den lugnande tolkningen att utvecklarna måste ha vunnit något annat — lärt sig mer, varit gladare eller skrivit bättre kod. Det som mättes här, den upplevda tidsvinsten, är precis det som data motsäger.

Hur starka är beläggen

- **Upplägget är ovanligt ärligt.** Randomiserat, verkliga uppgifter, verkliga kodförråd, faktisk tidtagning — ett stort steg framåt jämfört med de självrapporter och topplistor från standardiserade tester som de flesta påståenden om AI-kodning vilar på. Tidsökningen på 19% klarade författarnas robusthetskontroller.
- **Det mest överförbara resultatet är uppfattningsgapet.** Experternas intuition om den egna tidsvinsten med AI var fel med ungefär 40 procentenheter, åt det optimistiska hållet. Det är en varning om *varje* självrapporterad produktivitetsvinst med AI — inklusive siffrorna från den här studien.
- **Det är en ögonblicksbild, inte en trend.** METR:s egen uppföljning från februari 2026, med samma typ av utvecklare och nyare verktyg, pekar mot en tidsvinst — mycket grovt –18% för återkommande utvecklare och –4% för nya — men författarna betecknar det som svaga belägg, snedvridna av vilka som var villiga att delta (utvecklare avböjde i allt högre grad att arbeta utan AI, ersättningen sänktes och urvalet av uppgifter blev skevt). Den ärliga tolkningen är att bilden förändras och att även förändringen rapporteras utan att någon tumme läggs på vågen.

Varför det spelar roll

Merparten av debatten om AI och programmering bygger på demonstrationer och magkänsla: en elegant skärminspelning, ett självsäkert påstående, en avfärdande himling med ögonen. Det sällsynta, och det som gör detta läsvärt, är att någon genomförde det tråkiga experimentet — lotta, ta tid på verkligt arbete och sedan fråga människor hur det gick. Svaret är obekvämt för båda lägren. Det punkterar berättelsen om att AI alltid ger erfarna utvecklare turbofart i svår, välbekant kod. Och det punkterar den prydliga motsatsen — ”AI gör utvecklare långsammare, bevisat” — eftersom samma teams nyare data redan lutar åt andra hållet. Den mest bestående lärdomen är också den minsta och mest mänskliga: de som utförde arbetet kände sig snabbare medan de mätbart var långsammare. ”Det känns snabbare” är inte ett belägg för att det är det. Mät det.

Kort sammanfattning

I en randomiserad kontrollerad studie lät METR sexton erfarna utvecklare inom öppen källkod slutföra 246 verkliga uppgifter i kodbasen som de kände väl, med AI-verktyg från början av 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) tillåtna för en slumpmässigt vald hälft. Utvecklarna väntade sig att AI skulle minska tiden per uppgift med omkring 24%; i stället ökade tiden med 19% — och efteråt trodde de fortfarande att AI hade gjort dem omkring 20% snabbare. Detta uppfattningsgap är studiens skarpa, robusta kärna. Men det gäller sexton utvecklare, en enda snäv situation och en fixerad ögonblicksbild från början av 2025; författarna är tydliga med att det inte visar att AI misslyckas med att hjälpa de flesta utvecklare, och deras egen uppföljning från 2026 pekar redan mot en tidsvinst, med egna förbehåll. En noggrann mätning värd att ta på allvar — inte en dom över AI-kodning.

Källor

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/>