The Clean Paper

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

Pieredzējuši izstrādātāji ar MI jutās ātrāki, lai gan izmērāmi strādāja lēnāk — īstais secinājums un tas, ko šis pētījums nesaka par MI programmēšanā

METR nejaušinātā kontrolētā pētījumā 16 pieredzējuši atvērtā pirmkoda izstrādātāji veica 246 reālus uzdevumus labi pazīstamās kodu bāzēs, nejauši atļaujot izmantot 2025. gada sākuma MI rīkus (Cursor Pro ar Claude 3.5/3.7 Sonnet). Viņi gaidīja aptuveni 24% laika ietaupījumu; tā vietā uzdevumi prasīja par 19% vairāk laika, bet pēc darba viņi joprojām uzskatīja, ka MI viņus paātrinājis par aptuveni 20%. Šī uztveres plaisa ir pētījuma robustākais rezultāts. Tomēr tie ir tikai 16 izstrādātāji vienā šaurā kontekstā un fiksēts 2025. gada sākuma momentuzņēmums; autori uzsver, ka tas neparāda, ka MI nepalīdz lielākajai daļai izstrādātāju, un viņu pašu 2026. gada dati jau sliecas paātrinājuma virzienā, lai gan ar atrunām.

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

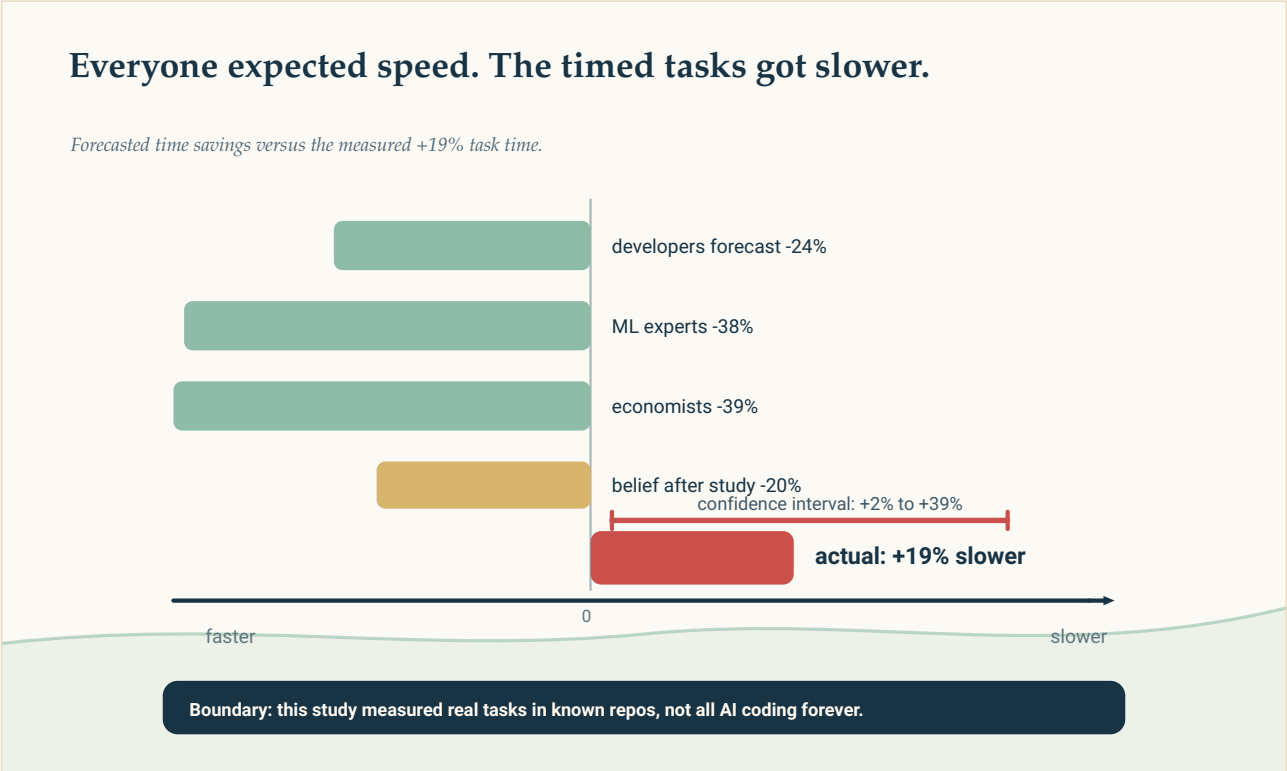
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Pieredzējuši izstrādātāji ar MI jutās ātrāki, lai gan izmērāmi strādāja lēnāk — īstais secinājums un tas, ko šis pētījums nesaka par MI programmēšanā

Pajautājiet pieredzējušam programmētājam, vai MI programmēšanas palīgs viņu paātrina, un parasti saņemsiet skaitli: ietaupa divdesmit, trīsdesmit procentus laika. Pajautājiet ekonomistam vai mašīnmācīšanās pētniekam, un skaitlis kļūs vēl lielāks. 2025. gada sākumā METR komanda izdarīja lēno un dārgo lietu — pārbaudīja. Tā iesaistīja sešpadsmit pieredzējušus atvērtā pirmkoda izstrādātājus, deva viņiem 246 reālus uzdevumus no lielām kodu bāzēm, kuras viņi labi pazina, un katram uzdevumam pēc nejaušības principa atļāva vai aizliedza izmantot MI rīkus. Pēc tam izmērīja darba laiku.

Izstrādātāji bija prognozējuši, ka MI saīsinās uzdevumu izpildes laiku par aptuveni 24%. Notika pretējais: uzdevumi, kuros drīkstēja izmantot MI, prasīja **par 19% vairāk laika**. Un te ir pētījuma interesantākā daļa — pēc darba pabeigšanas tie paši izstrādātāji joprojām uzskatīja, ka MI viņus bija paātrinājis par aptuveni 20%. Viņi strādāja lēnāk, bet jutās ātrāki; tieši plaisa starp šiem diviem skaitļiem ir pētījuma spilgtākais rezultāts.

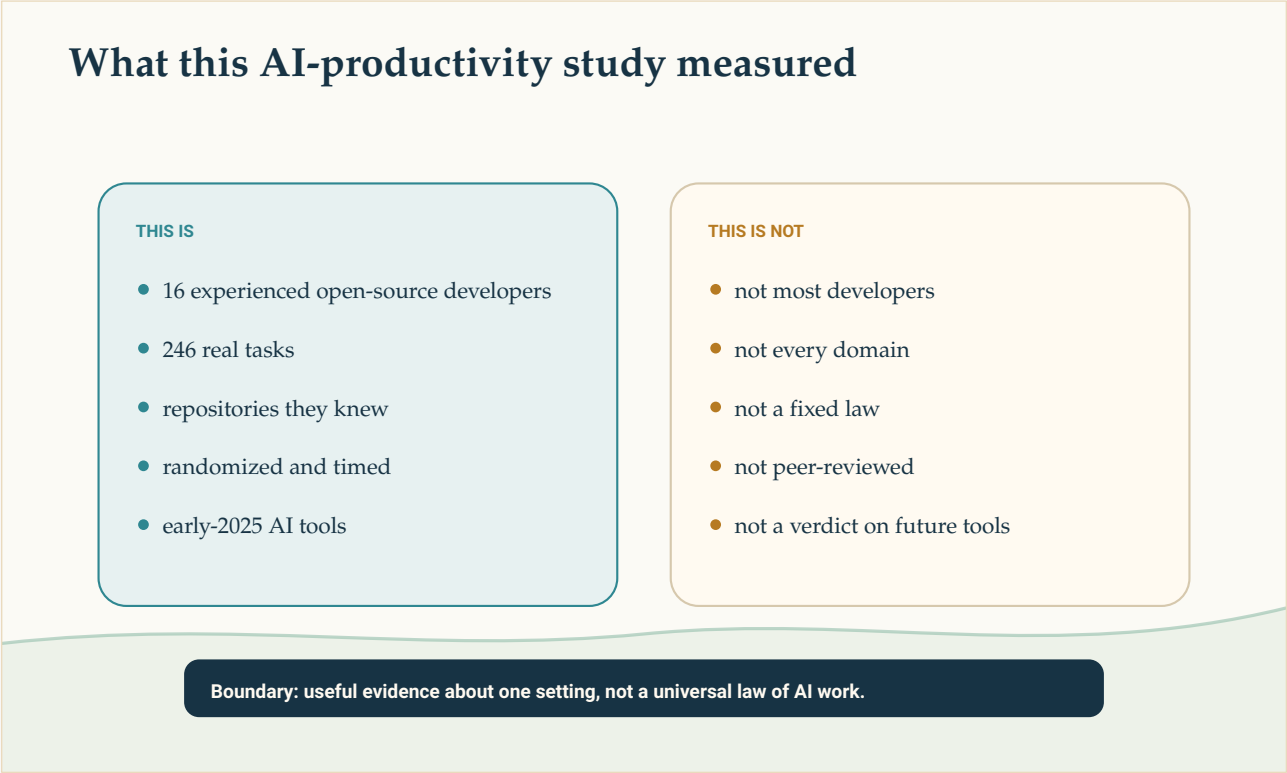
Tas ir reāls un rūpīgi izmērīts efekts. Taču runa ir arī tikai par sešpadsmit izstrādātājiem, par repozitorijiem, kurus viņi ļoti labi pazīst, un par 2025. gada sākuma rīkiem. Tas nav vienkāršs spriedums "MI padara izstrādātājus lēnākus". Tās pašas komandas jaunāki dati jau rāda citā virzienā.



Visas grupas paredzēja, ka MI paātrinās darbu — izstrādātāji par 24%, mašīnmācīšanās eksperti par 38%, ekonomisti par 39%, bet izstrādātāju pašu vērtējums pēc eksperimenta bija 20% paātrinājums. Hronometrs

uzrādīja pretējo: darbs bija par 19% lēnāks, ar intervālu no +2% līdz +39%. Atšķirība starp izjūtu un mērījumu ir galvenais rezultāts.

Original diagram — The Clean Paper CC BY 4.0



Ko pētījums mēra — 16 pieredzējušus atvērtā pirmkoda izstrādātājus, 246 reālus uzdevumus, pazīstamus repozitorijus, 2025. gada sākuma rīkus un nejaušinātu dizainu — un ko tas nemēra: ne lielāko daļu izstrādātāju, ne citas jomas, ne nemainīgu likumu; pētījums arī vēl nav recenzēts.

Original diagram — The Clean Paper CC BY 4.0

Ko te īsti izmērīja un ko dod nejaušināts pētījums

Nejaušināts kontrolēts pētījums (RCT) ir metode, ko medicīnā izmanto, lai nošķirtu reālu efektu no cerīga iespaida. Šeit katrs no 246 uzdevumiem pēc nejaušības principa nonāca grupā, kur MI bija atļauts, vai grupā, kur tas nebija atļauts. Tādēļ vidēji vienīgā sistemātiskā atšķirība starp abām grupām bija pats MI. Tas ļauj teikt, ka MI *izraisīja* laika izmaiņu, nevis tikai pamanīt, ka cilvēki, kuri izvēlas MI, citu iemeslu dēļ mēdz strādāt ātrāk vai lēnāk. Tas ir svarīgi, jo parastie pierādījumi par ieguvumiem no MI programmēšanā — pašvērtējumi un etalontestu rezultāti — to nespēj: etalontests nav reāls darbs, un pašvērtējums, kā rāda šis pētījums, var būt pārliecinoši kļūdains. Šeit nebija iesācēju, kuri tikai mēģina jaunu rotaļlietu. Tie bija regulāri liela, nobrieduša atvērtā pirmkoda projekta dalībnieki, kuri savus projektus labi pārzināja un kuriem jau bija zināma pieredze ar MI rīkiem.

Ko autori darīja

- Veica **nejaušinātu kontrolētu pētījumu** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Tajā piedalījās 16 pieredzējuši atvērtā pirmkoda izstrādātāji, kuri katrs strādāja pie liela repozitorija, kurā regulāri veic ieguldījumus un kuru labi pazīst — konkrētajos projektos vidēji piecus gadus.
- Izmantoja **246 reālus uzdevumus** — kļūdu labojumus, jaunas funkcijas un refaktorizāciju no pašu projektu uzdevumu sistēmām. Katram uzdevumam nejauši noteica "MI atļauts" vai "MI aizliegts".
- "MI atļauts" nozīmēja **2025. gada sākuma rīkus: Cursor Pro ar Claude 3.5/3.7 Sonnet**. Galvenais rādītājs bija faktiskais uzdevuma pabeigšanas laiks. Papildus komanda apkopoja prognozes pirms darba un novērtējumus pēc tā, kā arī ekonomikas un mašīnmācīšanās ekspertu paredzējumus.

Ko viņi atklāja

- **Ar MI uzdevumi prasīja par 19% vairāk laika.** Nevis mazāk, bet vairāk. 95% ticamības intervāls ir aptuveni no +2% līdz +39%, tātad efekta virziens ir samērā drošs, pat ja precīzs lielums nav.
- **Visi bija paredzējuši pretējo.** Izstrādātāji gaidīja 24% paātrinājumu, mašīnmācīšanās eksperti ap 38%, ekonomisti ap 39%. Visas trīs grupas gaidīja lielu laika ietaupījumu; hronometrs parādīja zināmu laika zaudējumu.
- **Uztveres plaisa.** Pēc darba, lai gan faktiski bija strādājuši lēnāk, izstrādātāji joprojām lēsa, ka MI viņus paātrinājis par aptuveni 20% — ap 40 procentpunktu atšķirība starp sajūtu un mērījumu.
- **Iespējamie iemesli ir apsvērti, nevis pierādīti.** Autori min vairākus faktorus, kas varētu izskaidrot palēninājumu: šie izstrādātāji ļoti labi pārzina savas kodu bāzes, tāpēc palīgam ir mazāk ko pievienot; nobriedušiem projektiem ir augsti, bieži neizteikti kvalitātes standarti; repozitoriji ir lieli un pilni ar kontekstu, kas modelim nav pieejams; un reāls laiks tiek patērēts uzvedņu rakstīšanai, pēc tam MI rezultātu pārbaudei un labošanai. Tie ir iespējamie skaidrojumi, nevis galīgi secinājumi.

Ko tas nepierāda

- Tas **neparāda**, ka MI nespēj paātrināt lielāko daļu izstrādātāju. Autori to saka tieši: sešpadsmit eksperti, kas kodu pazīst gandrīz no galvas, nav vidējais izstrādātājs vidējā uzdevumā.
- Tas **neparāda**, ka MI ir nederīgs vai ka tas palēnina cilvēkus citos apstākļos — piemēram, jaunpienācējus kodu bāzē, darbu no nulles, nepazīstamas programmēšanas valodas vai pavisam citas jomas.
- Tas **neiesaldē rīkus laikā**. Šeit runa ir par 2025. gada sākuma Cursor un Claude 3.5/3.7 Sonnet; autori skaidri norāda, ka labāki rīki vai labāks to lietojums varētu mainīt rezultātu pat tieši šādā pašā vidē.
- Tas ir **preprints** (publicēts 2025. gada jūlijā, vēl nav recenzēts), un autori norāda, ka nevar pilnībā izslēgt eksperimenta artefaktus, lai gan rezultāts saglabājās dažādās analīzēs.

- Tas arī **neļauj** mierinoši pieņemt, ka izstrādātāji varbūt ieguva kaut ko citu — vairāk iemācījās, jutās labāk vai uzrakstīja kvalitatīvāku kodu. Tieši tas, kas šeit tika mērīts kā sajūta — šķietamais paātrinājums — ir pretrunā ar datiem.

Cik pārliecinoši ir pierādījumi?

- **Pētījuma dizains ir neparasti tiešs.** Nejaušināšana, reāli uzdevumi, reāli repozitoriji un faktisks laika mērījums ir liels solis uz priekšu salīdzinājumā ar pašvērtējumiem un etalontestu līderu tabulām, uz kurām balstās daudzi apgalvojumi par MI programmēšanā. 19% palēninājums saglabājās autoru robustuma pārbaudēs.
- **Vispārināmākais secinājums ir uztveres plaisa.** Ekspertu intuīcija par savu paātrinājumu ar MI kļūdījās par aptuveni 40 procentpunktiem optimistiskā virzienā. Tas ir brīdinājums pret *jebkuru* pašziņotu produktivitātes pieaugumu no MI — arī pret šā pētījuma pašu pašvērtējumiem.
- **Tas ir momentuzņēmums, nevis tendence.** METR 2026. gada februāra turpinājums ar līdzīga tipa izstrādātājiem un jaunākiem rīkiem norāda uz paātrinājumu — ļoti aptuveni −18% tiem, kuri piedalījās atkārtoti, un −4% jaunajiem dalībniekiem. Tomēr autori šo pierādījumu sauc par vāju, jo atlasi ietekmēja tas, kuri bija gatavi piedalīties: izstrādātāji arvien biežāk atteicās strādāt bez MI, samaksa bija zemāka un mainījās uzdevumu atlase. Godīgais secinājums ir, ka aina mainās, un arī šo izmaiņu autori apraksta piesardzīgi.

Kāpēc tas ir svarīgi

Liela daļa strīdu par MI un programmēšanu balstās uz demonstrācijām un sajūtām: glīts ekrāna ieraksts, pārliecinošs apgalvojums, skeptiska reakcija. Retums — un iemesls, kāpēc šo pētījumu vērts lasīt — ir tas, ka kāds veica garlaicīgo eksperimentu: nejaušināja, izmērīja reālu darbu un pēc tam pajautāja cilvēkiem, kā viņiem šķita. Atbilde ir neērta abām nometnēm. Tā apstrīd stāstu, ka MI vienmēr dramatiski paātrina ekspertus sarežģītā, pazīstamā kodā. Un tā apstrīd arī gludo pretējo stāstu — "pierādīts, ka MI padara izstrādātājus lēnākus" — jo tās pašas komandas jaunākie dati jau sliecas citā virzienā. Izturīgākā mācība ir arī visvienkāršākā un cilvēciskākā: cilvēki jutās ātrāki, lai gan izmērāmi strādāja lēnāk. "Man šķiet, ka ir ātrāk" nav pierādījums. Tas ir jāizmēra.

Īss kopsavilkums

Nejaušinātā kontrolētā pētījumā METR lika 16 pieredzējušiem atvērtā pirmkoda izstrādātājiem paveikt 246 reālus uzdevumus kodu bāzēs, kuras viņi labi pazina; nejauši pusei uzdevumu drīkstēja izmantot 2025. gada sākuma MI rīkus (Cursor Pro kopā ar Claude 3.5/3.7 Sonnet). Izstrādātāji gaidīja, ka MI saīsinās darba laiku par aptuveni 24%; faktiski tas pagarināja uzdevumu izpildes laiku par 19% — un pēc tam viņi joprojām uzskatīja, ka MI viņus bija paātrinājis par aptuveni 20%. Šī uztveres plaisa ir pētījuma asākais un robustākais rezultāts. Taču tas ir tikai 16 izstrādātāju, šaurs konteksts un

fiksēts 2025. gada sākuma momentuzņēmums; autori skaidri norāda, ka pētījums neparāda, ka MI nepalīdz vairumam izstrādātāju, un viņu pašu 2026. gada turpinājums jau norāda uz paātrinājumu, lai gan ar būtiskām atrunām. Rūpīgs mērījums, ko vērts uztvert nopietni — nevis spriedums par MI programmēšanu.

Avoti

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/