



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

Patyrusiems programuotojams atrodė, kad su DI jie dirba greičiau, nors išmatuotas darbas truko ilgiau — štai tikrasis rezultatas, o ne galutinis nuosprendis DI programavimui

Atsitiktinių imčių kontroliuojamame METR tyrime šešiolika patyrusių atvirojo kodo programuotojų atliko 246 realias užduotis gerai pažįstamose kodų bazėse; atsitiktinei pusei užduočių buvo leidžiama naudoti 2025 m. pradžios DI įrankius (Cursor Pro ir Claude 3.5/3.7 Sonnet). Jie tikėjosi, kad DI sutrumpins darbą maždaug 24 %, tačiau iš tikrųjų užduočių atlikimo laikas padidėjo 19 % — ir po darbo jie vis tiek manė, kad DI juos paspartino apie 20 %. Būtent šis suvokimo atotrūkis yra tvirčiausias tyrimo rezultatas. Tačiau tai šešiolika programuotojų, siaura situacija ir vieno laikotarpio vaizdas; autoriai pabrėžia, kad tyrimas neparodo, jog DI nepadeda daugumai programuotojų, o jų pačių 2026 m. testiniai duomenys jau krypsta į pagreitėjimą, nors ir su savais apribojimais.

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

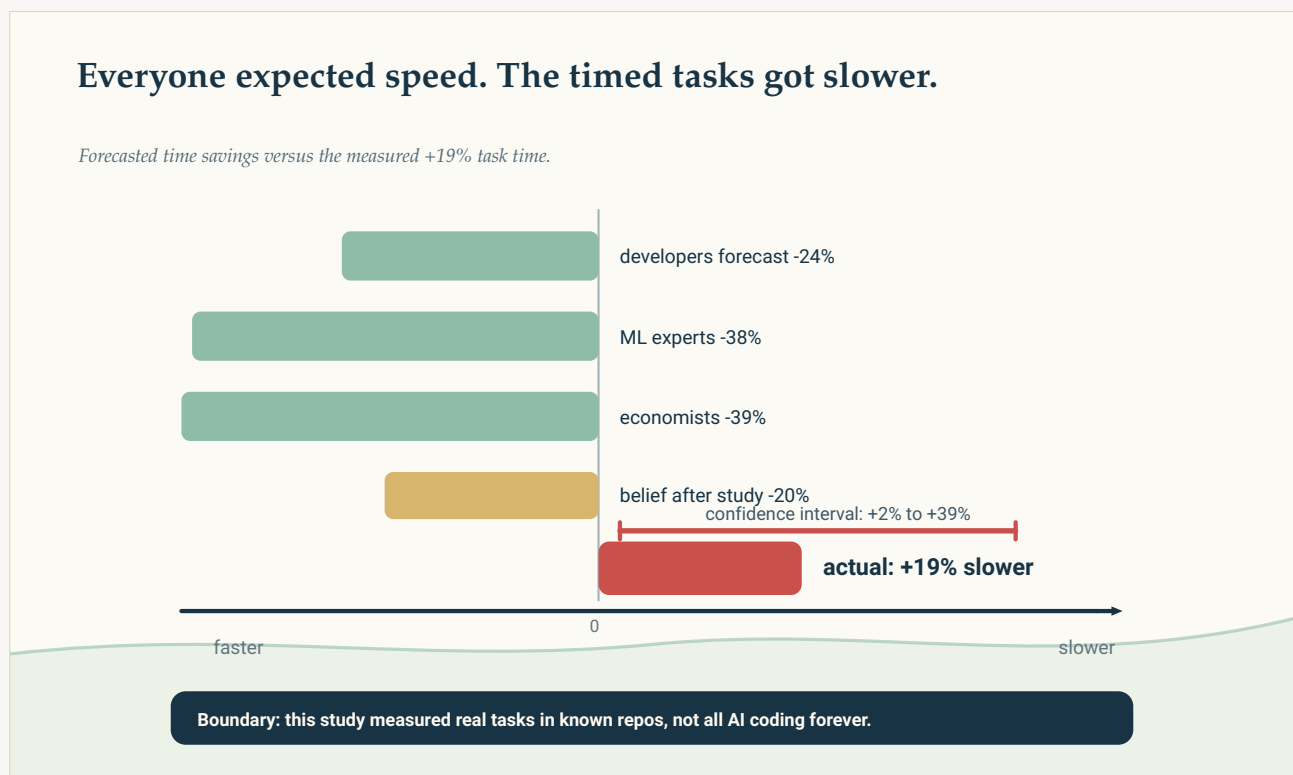
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Patyrusiems programuotojams atrodė, kad su DI jie dirba greičiau, nors išmatuotas darbas truko ilgiau — štai tikrasis rezultatas, o ne galutinis nuosprendis DI programavimui

Paklauskite patyrusio programuotojo, ar DI programavimo asistentas jį pagreitina, ir dažnai išgirsite skaičių: sutaupo dvidešimt, trisdešimt procentų laiko. Paklauskite ekonomisto ar mašininio mokymosi tyrėjo — prognozė paprastai bus dar didesnė. 2025 m. pradžioje METR komanda pasirinko lėtą ir brangų būdą tai patikrinti. Ji subūrė šešiolika patyrusių atvirojo kodo programuotojų, pateikė jiems 246 tikras užduotis iš didelių, gerai pažįstamų kodų bazių ir kiekvienai užduočiai atsitiktinai leido arba neleido naudoti DI įrankių. Tada tiesiog išmatavo, kiek truko darbas.

Programuotojai iš anksto prognozavo, kad DI užduočių laiką sutrumpins maždaug 24 %. Įvyko priešingai: su DI atliktos užduotys truko **19 % ilgiau**. Ir čia yra verta sustoti: baigę darbą tie patys programuotojai vis tiek manė, kad DI juos paspartino maždaug 20 %. Jie dirbo lėčiau, bet jautėsi dirbę greičiau, o atstumas tarp šių dviejų skaičių ir yra įdomiausia tyrimo dalis.

Tai tikras ir kruopščiai išmatuotas rezultatas. Tačiau kalbame apie šešiolika programuotojų, dirbusių su sau labai gerai pažįstamomis repozitorijomis ir naudojusių 2025 m. pradžios įrankius. Iš to negalima daryti paprastos išvados „DI lėtina programuotojus“. Tos pačios komandos vėlesni duomenys jau rodo galimą posūkį į kitą pusę.



Visos grupės prognozavo, kad DI darbą pagreitins: programuotojai — 24 %, mašininio mokymosi ekspertai — 38 %, ekonomistai — 39 %, o pačių programuotojų vertinimas po tyrimo — 20 %. Laiko matavimas parodė priešingai: darbas buvo 19 % lėtesnis, o pasikliautinis intervalas siekė nuo +2 % iki +39 %. Skirtumas tarp to, kaip žmonės jautėsi, ir to, kas buvo išmatuota, yra svarbiausias rezultatas.

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.

Ką tyrimas matavo: 16 patyrusių atvirojo kodo programuotojų, 246 realias užduotis, gerai pažįstamas repozitorijas, 2025 m. pradžios įrankius ir atsitiktinį paskirstymą. Ko jis nematavo: daugumos programuotojų, kitų sričių, nekintamo dėsno ar recenzuoto galutinio atsakymo.

Original diagram — The Clean Paper CC BY 4.0

Ką čia iš tiesų matavo ir kuo naudingas atsitiktinių imčių tyrimas

Atsitiktinių imčių kontroliuojamas tyrimas (RCT) yra metodas, kuriuo medicinoje ir kitose srityse stengiamasi atskirti tikrą poveikį nuo viltingo įspūdžio. Čia kiekviena iš 246 užduočių atsitiktinai pateko į vieną iš dviejų sąlygų: DI naudoti leidžiama arba ne. Todėl vidutiniškai vienintelis sistemingas skirtumas tarp abiejų grupių turėtų būti pats DI. Būtent tai leidžia teigti, kad DI *sukėlė* laiko pokytį, o ne vien pastebėti, jog žmonės, kurie dažniau griebiasi DI, dėl kitų priežasčių būna greitesni ar lėtesni. Tai svarbu todėl, kad dažniausi argumentai apie DI programavimo naudą — savęs vertinimai ir testų rezultatai — tokio priežastinio ryšio neparodo: testas nėra tikras darbas, o savęs vertinimas, kaip rodo šis tyrimas, gali būti užtikrintai klaidingas. Tyrimo dalyviai nebuvo naujokai, pirmą kartą bandantys naują žaislą. Jie buvo patyrę didelių, brandžių atvirojo kodo projektų bendradarbiai, gerai pažįstantys savo kodą ir jau turėję šiek tiek patirties su DI įrankiais.

Ką padarė autoriai

- Atliko **atsitiktinių imčių kontroliuojamą tyrimą** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Dalyvavo šešiolika patyrusių atvirojo kodo programuotojų; kiekvienas dirbo su didele repozitorija, prie kurios nuolat prisideda ir kurią gerai pažįsta — vidutiniškai konkrečiuose projektuose jie buvo dirbę penkerius metus.
- Naudojo **246 realias užduotis** — klaidų taisymus, naujas funkcijas ir kodo pertvarkymus iš pačių projektų užduočių sistemų. Kiekviena užduotis atsitiktinai priskirta sąlygai „DI leidžiamas“ arba „DI neleidžiamas“.
- „DI leidžiamas“ reiškė **2025 m. pradžios įrankius: Cursor Pro su Claude 3.5/3.7 Sonnet**. Pagrindinis rodiklis buvo tikrasis kiekvienos užduoties atlikimo laikas. Be to, komanda rinko dalyvių prognozes prieš darbą ir vertinimus po jo, taip pat ekonomikos bei mašininio mokymosi ekspertų prognozes.

Ką jie nustatė

- **Su DI užduotys truko 19 % ilgiau.** Ne trumpiau, o ilgiau. 95 % pasikliautinis intervalas apėmė maždaug nuo +2 % iki +39 %, todėl poveikio kryptis gana aiški, nors tikslus dydis nėra toks tvirtas.
- **Visi prognozavo priešingai.** Programuotojai tikėjosi 24 % pagreitėjimo, mašininio mokymosi ekspertai — apie 38 %, ekonomistai — apie 39 %. Visos trys grupės manė, kad DI sutaupys daug laiko; matavimas parodė, kad jo prireikė daugiau.
- **Suvokimo atotrūkis.** Net atlikę darbus lėčiau programuotojai po tyrimo vis tiek vertino, kad DI juos paspartino maždaug 20 %. Tai maždaug 40 procentinių punktų skirtumas tarp pojūčio ir chronometro.
- **Galimos priežastys įvertintos, bet neįrodytos.** Autoriai aptaria kelis lėtėjimo paaiškinimus: šie programuotojai labai gerai pažįsta savo kodų bazes, todėl asistentas turi mažiau kuo padėti; brandūs projektai turi aukštus, dažnai neišsakytus kokybės reikalavimus; repozitorijos didelės ir pilnos konteksto, kurio modelis neturi; be to, realus laikas išeina užklausoms rašyti, DI atsakymams peržiūrėti ir taisyti. Tai pateikiama kaip galimos kryptys, o ne kaip įrodyti verdiktai.

Ko tai neparodo

- Jis **neparodo**, kad DI nepagreitina daugumos programuotojų. Autoriai tai sako tiesiai: šešiolika ekspertų, dirbančių su beveik mintinai pažįstamu kodu, nėra vidutinis programuotojas ir vidutinė užduotis.
- Jis **neparodo**, kad DI yra nenaudingas ar kad kitomis sąlygomis žmonės lėtina — pavyzdžiui, naujokams projekte, kuriant naują sistemą nuo nulio, dirbant nepažįstama kalba ar visai kitoje srityje.
- Jis **neužfiksuoja įrankių visiems laikams**. Čia naudotas 2025 m. pradžios Cursor ir Claude 3.5/3.7 Sonnet. Autoriai aiškiai pripažįsta, kad geresni įrankiai arba geresni jų naudojimo būdai gali pakeisti rezultatą net tokioje pačioje situacijoje.

- Tai **preprintas** (paskelbtas 2025 m. liepą, dar nerecenzuotas), o autoriai pažymi negalintys visiškai atmesti eksperimentinių artefaktų — nors rezultatas išliko panašus skirtingose analizėse.
- Iš tyrimo taip pat **negalima guostis**, esą programuotojai būtinai laimėjo kitu būdu — daugiau išmoko, buvo laimingesni ar parašė geresnį kodą. Vienas dalykas, kuris čia tikrai buvo matuojamas — jaučiamas pagreitėjimas — būtent ir prieštarauja duomenims.

Kiek tvirti yra įrodymai

- **Tyrimo dizainas neįprastai tiesus.** Atsitiktinis paskirstymas, realios užduotys, realios repozitorijos ir tikras laiko matavimas yra didelis žingsnis pirmyn nuo savęs vertinimų ir testų reitingų, kuriais remiasi daug teiginių apie DI programavimą. 19 % sulėtėjimas išliko ir autorių patikimumo patikrose.
- **Plačiausiai pritaikomas rezultatas yra suvokimo atotrūkis.** Ekspertų intuicija apie jų pačių darbo pagreitėjimą buvo klaidinga maždaug 40 procentinių punktų ir klaida buvo optimistinė. Tai perspėjimas atsargiai vertinti *visus* savęs vertinimu pagrįstus DI produktyvumo skaičius — įskaitant ir šio tyrimo dalyvių nuomonę.
- **Tai vieno laikotarpio vaizdas, o ne ilgalaikė tendencija.** METR 2026 m. vasario testinis tyrimas su panašaus profilio programuotojais ir naujesniais įrankiais jau krypta į pagreitėjimą — labai apytikriai apie –18 % grįžusiems dalyviams ir –4 % naujiems. Tačiau autoriai patys tai vadina silpnais duomenimis, nes juos iškraipė atranka: programuotojai vis dažniau atsisakydavo dalyvauti, jei negalėjo naudoti DI, sumažėjo atlygis ir pakito užduočių pasirinkimas. Sąžininga išvada tokia: padėtis juda, bet ir patį judėjimą reikia matuoti atsargiai.

Kodėl tai svarbu

Didelė dalis ginčo apie DI ir programavimą remiasi demonstracijomis ir nuojauta: sklandžiu vaizdo įrašu, užtikrintu teiginiu ir skeptišku atsakymu. Retas dalykas, dėl kurio šį darbą verta skaityti, yra nuobodus eksperimentas: atsitiktinai paskirstyti sąlygas, pamatuoti tikrą darbą, o tada paklausti dalyvių, kaip jiems sekėsi. Atsakymas nepatogus abiem stovykloms. Jis praduria pasakojimą, kad DI vienodai stipriai pagreitina patyrusius programuotojus, dirbančius su sudėtingu ir gerai pažįstamu kodu. Bet taip pat neleidžia patogiai skelbti priešingos išvados — „įrodyta, kad DI lėtina programuotojus“ — nes naujesni tos pačios komandos duomenys jau krypta kita kryptimi. Patvariausia pamoka yra ir mažiausia, ir žmogiškiausia: žmonės jautėsi dirbę greičiau, nors išmatuotai dirbo lėčiau. „Atrodo greičiau“ nėra įrodymas, kad taip ir yra. Reikia matuoti.

Trumpa santrauka

Atsitiktinių imčių kontroliuojamame tyrime METR paprašė šešiolikos patyrusių atvirojo kodo programuotojų atlikti 246 tikras užduotis jiems gerai pažįstamose kodų bazėse; atsitiktinei pusei užduočių

buvo leidžiama naudoti 2025 m. pradžios DI įrankius (Cursor Pro ir Claude 3.5/3.7 Sonnet). Programuotojai tikėjosi, kad DI sutrumpins darbą maždaug 24 %, tačiau iš tikrųjų užduočių atlikimo laikas padidėjo 19 % — ir po darbo jie vis tiek manė, kad DI juos paspartino maždaug 20 %. Šis suvokimo atotrūkis yra aiškiausias ir tvirčiausias tyrimo rezultatas. Tačiau tai tik šešiolika programuotojų, labai siaura situacija ir vienas 2025 m. pradžios momentas; autoriai aiškiai pabrėžia, kad tyrimas neparodo, jog DI nepadedą daugumai programuotojų, o jų pačių 2026 m. tęstiniai duomenys jau krypta į pagreitinimą, nors ir su savais apribojimais. Tai matavimas, kurį verta vertinti rimtai — ne galutinis nuosprendis DI programavimui.

Šaltiniai

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/>