The
CLEAN PAPER

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

A tapasztalt fejlesztők gyorsabbnak érezték magukat MI-vel, miközben mérhetően lassabban dolgoztak — ez a valódi eredmény, nem ítélet az MI-s kódolásról

A METR randomizált kontrollált vizsgálatában tizenhat tapasztalt nyílt forráskódú fejlesztő 246 valódi feladatot végzett jól ismert kódbázisokon; a feladatok véletlenszerű felében 2025 eleji MI-eszközöket használhattak (Cursor Pro + Claude 3.5/3.7 Sonnet). Azt várták, hogy az MI körülbelül 24%-kal csökkenti a feladatidőt; ehelyett 19%-kal növelte — utólag mégis úgy érezték, hogy mintegy 20%-kal gyorsabbak voltak. Ez az észlelési rés a vizsgálat legerősebb magja. De tizenhat fejlesztőről, szűk környezetről és rögzített 2025 eleji pillanatfelvételről van szó; a szerzők szerint ez nem bizonyítja, hogy az MI a fejlesztők többségének nem segít, saját 2026-os utánkövetésük pedig már gyorsulás felé hajlik, a maga korlátaival.

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

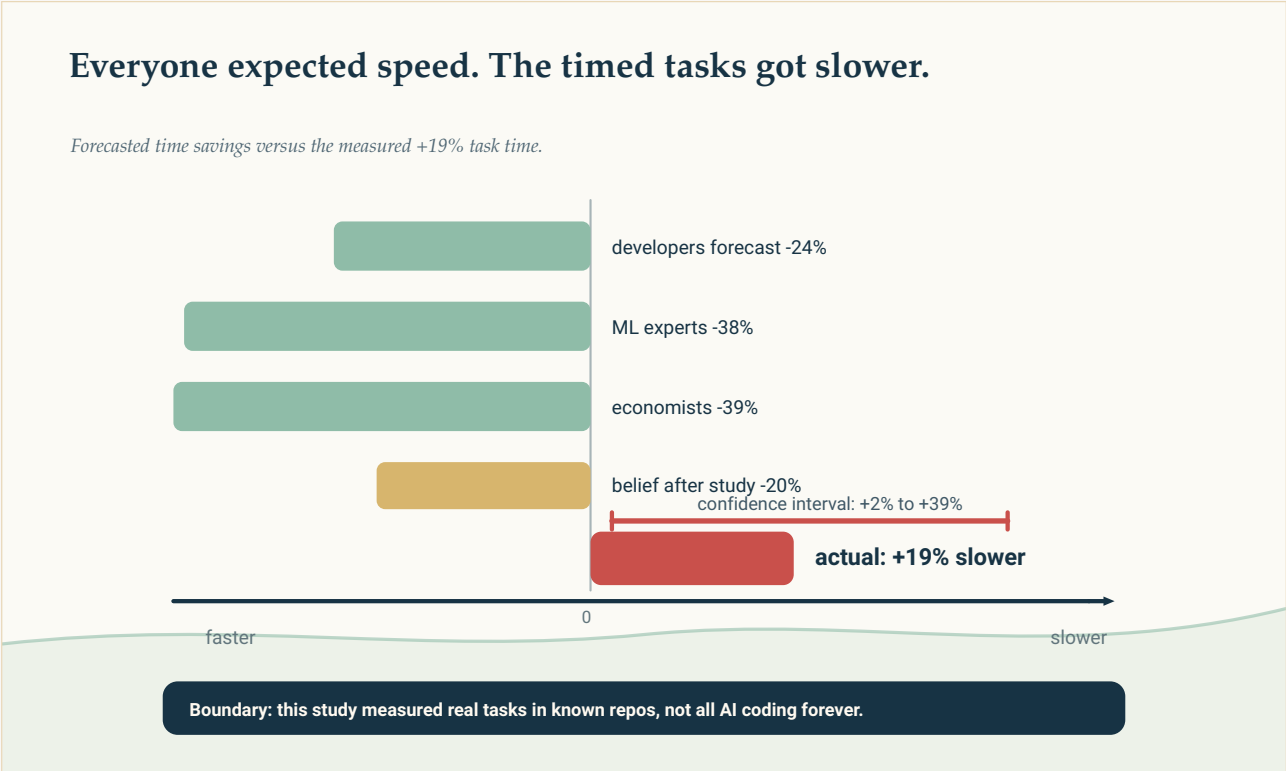
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

A tapasztalt fejlesztők gyorsabbnak érezték magukat MI-vel, miközben mérhetően lassabban dolgoztak — ez a valódi eredmény, nem pedig ítélet az MI-alapú programozásról

Ha megkérdezünk egy tapasztalt programozót, gyorsítja-e az MI-s kódolási asszisztens, általában kapunk egy számot: húsz, harminc százalék időt spórol. Ha közgazdászt vagy gépi tanulással foglalkozó kutatót kérdezünk, a szám még nagyobb lesz. 2025 elején a METR egyik csapata elvégezte a lassú és drága dolgot: ellenőrizte. Tizenhat tapasztalt nyílt forráskódú fejlesztőnek 246 valódi feladatot adtak olyan nagy kódbázisokból, amelyeket jól ismertek, majd feladatonként véletlenszerűen eldöntötték, használhatnak-e MI-eszközöket. Ezután megmérték az időt.

A fejlesztők előre azt becsülték, hogy az MI körülbelül 24%-kal csökkenti a feladatokra fordított időt. Az ellenkezője történt: az MI-vel végzett feladatok **19%-kal tovább tartottak**. És itt jön az a rész, amin érdemes elidőzni: a munka után ugyanazok a fejlesztők továbbra is úgy gondolták, hogy az MI körülbelül 20%-kal felgyorsította őket. Lassabbak voltak, miközben gyorsabbnak érezték magukat; a két szám közötti távolság a tanulmány legérdekesebb eredménye.

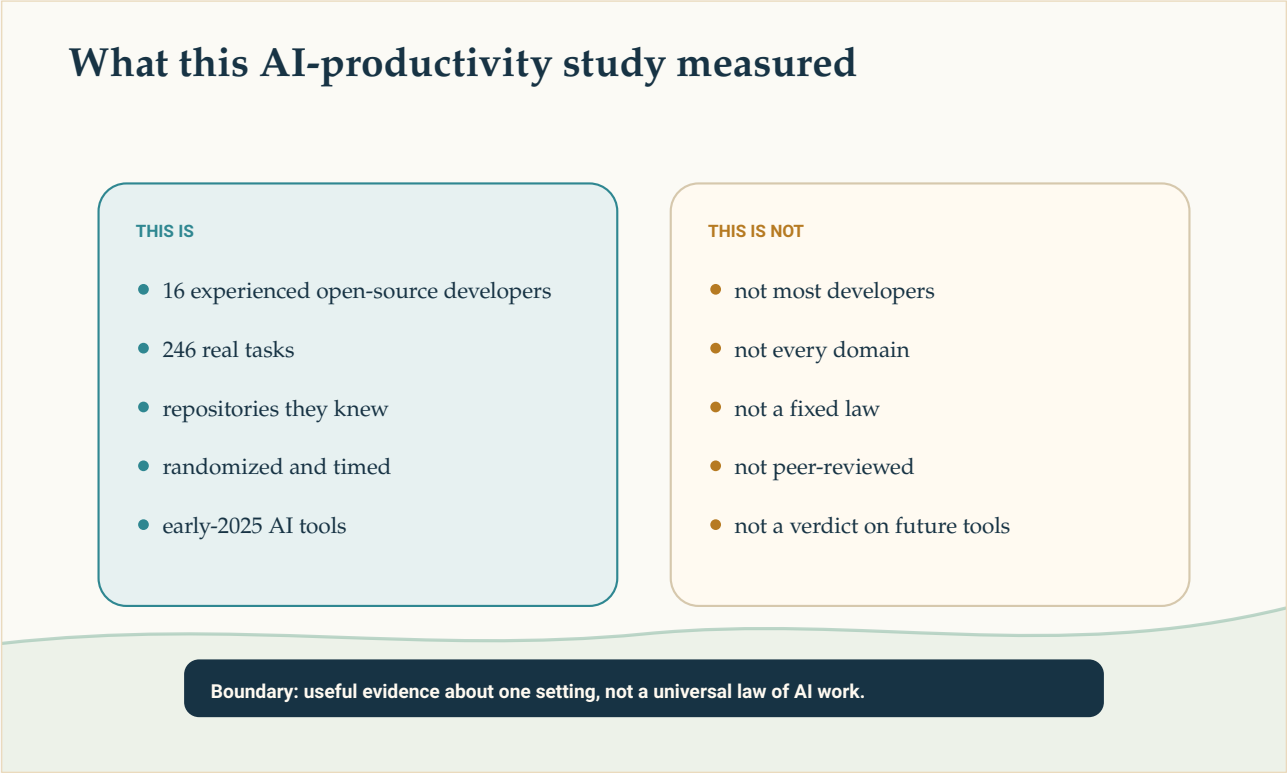
Ez valós, gondosan mért eredmény. Ugyanakkor tizenhat fejlesztőről szól, olyan repozitóriumokon, amelyeket rendkívül jól ismernek, 2025 eleji eszközökkel — és nem az a lapos mondat következik belőle, hogy „az MI lassabbá teszi a fejlesztőket”. Ugyanennek a csapatnak a későbbi adatai már az ellenkező irányba mutatnak.



Mindenki arra számított, hogy az MI gyorsítja a munkát — fejlesztők: −24%, gépi tanulási szakértők: −38%, közgazdászok: −39%, a fejlesztők utólagos saját becslése: −20%. A stopper ezzel szemben +19%-os lassulást

talált, +2% és +39% közötti intervallummal. Az érzékelt és a mért eredmény közötti rés maga a fontos eredmény.

Original diagram — The Clean Paper CC BY 4.0



Mit mér a vizsgálat: 16 szakértő nyílt forráskódú fejlesztő, 246 valódi feladat, jól ismert repozitóriumok, 2025 eleji eszközök, randomizálás. És mit nem: nem a fejlesztők többségét, nem más területeket, nem megváltoztathatatlan törvényt, és nem lektorált eredményt.

Original diagram — The Clean Paper CC BY 4.0

Mit mért ez a vizsgálat, és mit ad hozzá a randomizálás?

A **randomizált kontrollált vizsgálat (RCT)** az az eszköz, amelyet az orvostudomány is használ annak elkülönítésére, hogy egy hatás valódi-e vagy csak remélt. Itt a 246 feladat mindegyikét véletlenszerűen osztották „MI használható” vagy „MI nem használható” feltételbe, így átlagosan a két csoport közötti egyetlen szisztematikus különbség maga az MI. Ez teszi lehetővé annak állítását, hogy az MI *okozta* az idő változását, nem pedig azt látjuk, hogy azok, akik amúgy is MI-hez nyúlnak, valamilyen más okból gyorsabbak vagy lassabbak. Ez azért fontos, mert az MI-s programozás termelékenységnövekedésére vonatkozó szokásos bizonyíték — önbevallás és benchmarkpontszám — erre nem képes: a benchmark nem valódi munka, az önbevallás pedig, amint ez a tanulmány mutatja, magabiztosan téves lehet. Az itt részt vevő fejlesztők nem kezdők voltak, akik ügyetlenkednek egy új játékszerrel. Nagy, kiforrott nyílt forrású projektek régi közreműködői voltak, amelyeket jól ismertek, és valamennyi korábbi tapasztalatuk is volt az eszközökkel.

Mit csináltak a szerzők?

- **Randomizált kontrollált vizsgálatot** végeztek (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Tizenhat tapasztalt nyílt forráskódú fejlesztő dolgozott egy-egy nagy repozitóriumon, amelyhez rendszeresen hozzájárult és amelyet jól ismert — átlagosan öt éve dolgoztak az adott projekten.
- **246 valódi feladatot** használtak: hibajavításokat, új funkciókat és refaktorálásokat a projektek saját hibakövetőiből. Minden feladatot véletlenszerűen „MI engedélyezett" vagy „MI tiltott" feltételbe osztottak.
- Az „MI engedélyezett" **2025 eleji eszközöket jelentett: Cursor Pro Claude 3.5/3.7 Sonnet modellel.** Az elsődleges mérőszám a feladatonkénti tényleges befejezési idő volt. Emellett előzetes becsléseket kértek a fejlesztőktől, utólagos becsléseket a munka után, valamint közgazdasági és gépi tanulási szakértők előrejelzéseit.

Mit találtak?

- **MI-vel a feladatok 19%-kal tovább tartottak.** Nem gyorsabban, hanem lassabban. A 95%-os konfidenciaintervallum nagyjából +2% és +39% között van, tehát az irány meglehetősen biztos, még ha a pontos nagyság nem is.
- **Mindenki az ellenkezőjét jósolta.** A fejlesztők 24%-os gyorsulást vártak; a gépi tanulási szakértők körülbelül 38%-ot; a közgazdászok mintegy 39%-ot. Mindhárom csoport nagy időmegtakarítást várt az MI-től; a stopper szerint inkább időbe került.
- **Az észlelési rés.** Miután a fejlesztők a mérés szerint lassabban végezték a munkát, továbbra is úgy becsülték, hogy az MI körülbelül 20%-kal felgyorsította őket — nagyjából 40 pontos különbség aközött, amit éreztek, és amit az óra mért.
- **Lehetséges okok, mérlegelve, nem bizonyítva.** A szerzők több tényezőt sorolnak fel, amelyek magyarázhatják a lassulást: ezek a fejlesztők nagyon mélyen ismerik saját kódbázisaikat, ezért kevesebb hozzáadott értéket ad egy asszisztens; a kiforrott projektekben magas és gyakran implicit minőségi elvárások vannak; a repozitóriumok nagyok és sok olyan kontextust tartalmaznak, amelyhez a modell nem fér hozzá; és valódi idő megy el a promptolásra, majd az MI kimenetének ellenőrzésére és javítására. Ezeket nyomokként, nem végleges magyarázatként mutatják be.

Mit nem mutat meg ez?

- **Nem** mutatja, hogy az MI a fejlesztők többségét nem gyorsítja. A szerzők ezt közvetlenül kimondják: tizenhat szakértő, akik kívülről-belülről ismerik saját kódjukat, nem azonos az átlagos fejlesztővel és az átlagos feladattal.
- **Nem** mutatja, hogy az MI haszontalan vagy más helyzetekben is lassít — például újonnan érkezőknek egy kódbázisban, zöldmezős fejlesztésnél, ismeretlen nyelveken vagy teljesen más területeken.

- **Nem** fagyasztja be az eszközöket az időben. Ez 2025 eleji Cursor és Claude 3.5/3.7 Sonnet; a szerzők hangsúlyozzák, hogy jobb eszközök vagy ugyanazon eszközök jobb használata akár pontosan ebben a környezetben is megváltoztathatja az eredményt.
- Ez egy **preprint** (2025 júliusában tették közzé, még nem lektorálták), és a szerzők megjegyzik, hogy nem tudják teljesen kizárni a kísérleti műtermékeket — bár az eredmény különböző elemzéseikben fennmaradt.
- **Nem** jogosít fel arra a megnyugtató értelmezésre sem, hogy a fejlesztők biztosan valami másban nyertek — többet tanultak, boldogabbak voltak, jobb kódot írtak. Az itt ténylegesen mért dolog, az érzékelt gyorsulás éppen az, aminek az adatok ellentmondanak.

Mennyire erős a bizonyíték?

- **A vizsgálati terv szokatlanul tisztességes.** Randomizált, valódi feladatokkal, valódi repozitóriumokon, tényleges időméréssel — nagy előrelépés az önbevallásokhoz és benchmarkrangsorokhoz képest, amelyekre az MI-s programozási állítások nagy része épül. A 19%-os lassulás túlélte a szerzők robusztussági ellenőrzéseit.
- **A leginkább általánosítható eredmény az észlelési rés.** A szakértők saját MI-s gyorsulásukra vonatkozó intuíciója körülbelül 40 ponttal tévedett, optimista irányban. Ez figyelmeztetés *minden* önbevallott MI-termelékenységnövekedés kapcsán — beleértve ennek a tanulmánynak a résztvevőit is.
- **Pillanatfelvétel, nem trend.** A METR saját, 2026 februári utánkövetése hasonló fejlesztőkkel és újabb eszközökkel már gyorsulás felé mutat — nagyon nagyjából −18% a visszatérő fejlesztőknél és −4% az új résztvevőknél —, de a szerzők ezt gyenge bizonyítéknak minősítik, mert erősen torzíthatta, kik voltak hajlandók részt venni (a fejlesztők egyre gyakrabban utasították el az MI nélküli munkát, a díjazás csökkent, a feladatválasztás eltolódott). Az őszinte olvasat: a kép változik, és még ezt a változást is óvatosan jelentik.

Miért fontos?

Az MI-ről és programozásról szóló vita nagy része demókból és benyomásokból él: látványos képernyőfelvétel, magabiztos állítás, majd szemforgató ellenreakció. Ritka az, ami ezt a tanulmányt olvasásra érdemessé teszi: valaki elvégezte az unalmas kísérletet — randomizált, valódi munkát időzített, majd megkérdezte az embereket, hogyan ment. A válasz mindkét tábor számára kényelmetlen. Kilyukasztja azt a történetet, hogy az MI egyformán felturbózza a szakértő fejlesztőket nehéz, jól ismert kódon. De a rendezett ellenkezőjét is — „bizonyítottan lassítja a fejlesztőket” —, mert ugyanazon csapat újabb adatai már a másik irányba hajlanak. A legtartósabb tanulság egyben a legkisebb és legemberibb: a munkát végzők gyorsabbnak érezték magukat, miközben mérhetően lassabbak voltak. Az, hogy „gyorsabbnak érződik”, nem bizonyíték arra, hogy valóban az. Mérjük meg.

Röviden

Egy randomizált kontrollált vizsgálatban a METR tizenhat tapasztalt nyílt forráskódú fejlesztővel 246 valódi feladatot végeztetett olyan kódbázisokon, amelyeket jól ismertek; a feladatok véletlenszerű felében engedélyezték a 2025 eleji MI-eszközöket (Cursor Pro + Claude 3.5/3.7 Sonnet). A fejlesztők arra számítottak, hogy az MI körülbelül 24%-kal csökkenti a feladatidőt; ehelyett 19%-kal növelte — és utólag mégis úgy gondolták, hogy mintegy 20%-kal felgyorsította őket. Ez az észlelési rés a vizsgálat éles, robusztus magja. De mindössze tizenhat fejlesztőről, egy szűk környezetről és egy rögzített 2025 eleji pillanatfelvételről van szó; a szerzők kifejezetten jelzik, hogy ez nem mutatja, hogy az MI a fejlesztők többségének nem segít, saját 2026-os utánkövetésük pedig már gyorsulás felé mutat, a maga korlátaival. Gondosan mért eredmény, amelyet érdemes komolyan venni — nem ítélet az MI-alapú programozás fölött.

Források

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/