



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

Doświadczeni programiści czuli, że z AI pracują szybciej, choć pomiary wykazały spowolnienie — rzeczywisty wynik, ale nie werdykt o programowaniu z AI

W randomizowanym badaniu kontrolowanym METR zlecił szesnastu doświadczonym programistom open source wykonanie 246 rzeczywistych zadań w dobrze znanych im bazach kodu, losowo zezwalając na użycie narzędzi AI z początku 2025 roku (Cursor Pro oraz Claude 3.5/3.7 Sonnet) w połowie zadań. Oczekiwali, że AI skróci czas o około 24%; zamiast tego wydłużyła go o 19% — a później nadal sądzili, że przyspieszyła ich o około 20%. Ta luka w postrzeganiu jest wyraźnym i odpornym rdzeniem badania. Dotyczy ono jednak szesnastu programistów, jednego wąskiego środowiska i określonego momentu na początku 2025 roku: autorzy podkreślają, że nie pokazuje, iż AI nie pomaga większości programistów, a ich własne badanie uzupełniające z 2026 roku już wskazuje na przyspieszenie, choć ma własne zastrzeżenia.

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

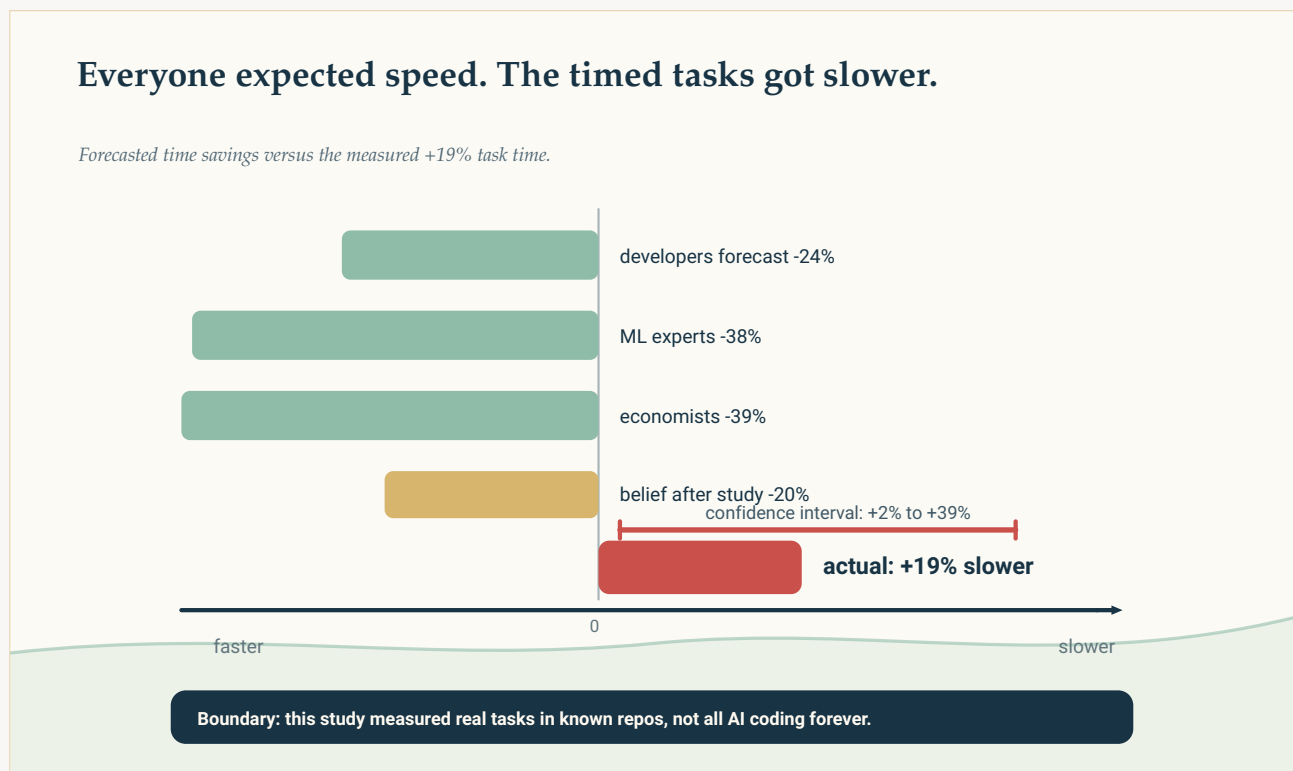
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Doświadczeni programiści czuli, że z AI pracują szybciej, choć pomiary wykazały spowolnienie — rzeczywisty wynik, ale nie werdykt o programowaniu z AI

Gdy zapytać doświadczonego programistę, czy asystent AI przyspiesza jego pracę, zwykle poda liczbę: oszczędza mi dwadzieścia, trzydzieści procent czasu. Gdy zapytać ekonomistę albo badacza uczenia maszynowego, liczba rośnie. Na początku 2025 roku zespół METR zrobił rzecz powolną i kosztowną: sprawdził to. Szesnastu doświadczonym twórcom oprogramowania open source przydzielono 246 rzeczywistych zadań z dużych baz kodu, które dobrze znali, a następnie — losowo, osobno dla każdego zadania — pozwalano im korzystać z narzędzi AI albo nie. Zmierzono czas pracy.

Programiści przewidywali, że AI skróci czas wykonania zadań o około 24%. Stało się odwrotnie: zadania wykonywane z AI zajęły **o 19% więcej czasu**. I właśnie nad kolejną częścią warto się zatrzymać — po zakończeniu pracy ci sami programiści nadal sądzili, że AI przyspieszyła ich o około 20%. Byli wolniejsi, a czuli się szybsi; różnica między tymi dwiema liczbami jest najciekawszym wynikiem badania.

To prawdziwy, starannie zmierzony rezultat. Dotyczy jednak szesnastu programistów pracujących w repozytoriach, które znają od podszewki, z narzędziami z początku 2025 roku — i nie sprowadza się do kategorycznego zdania „AI spowalnia programistów”. Późniejsze dane tego samego zespołu wskazują już w przeciwną stronę.



Wszyscy przewidywali, że AI przyspieszy pracę — programiści o 24%, eksperci uczenia maszynowego o 38%, ekonomiści o 39%, a sami programiści po fakcie nadal oceniali przyspieszenie na 20%. Stoper wykazał

spowolnienie o 19%, z przedziałem od 2% do 39%. Wynikiem jest różnica między odczuciem a pomiarem.

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.

Co mierzy badanie — 16 doświadczonych programistów open source, 246 rzeczywistych zadań, znane repozytoria, narzędzia z początku 2025 roku i przydział losowy — oraz czego nie mierzy: większości programistów, innych dziedzin, niezmiennego prawa ani wyniku po recenzji naukowej.

Original diagram — The Clean Paper CC BY 4.0

Co właściwie zmierzono i co daje badanie z randomizacją

Randomizowane badanie kontrolowane (RCT) jest narzędziem używanym w medycynie do odróżnienia rzeczywistego efektu od oczekiwanego. Tutaj każde z 246 zadań losowo przypisano do wykonania z dozwolonym AI albo bez niego, dzięki czemu jedyną systematyczną różnicą między obiema grupami jest przeciętnie samo AI. To pozwala powiedzieć, że AI *spowodowała* zmianę czasu, zamiast jedynie zauważyć, że osoby sięgające po AI z innych powodów bywają szybsze lub wolniejsze. Ma to znaczenie, ponieważ typowe dowody na korzyści z AI w programowaniu — samooceny i wyniki benchmarków — nie dają takiej możliwości: benchmark nie jest prawdziwą pracą, a samoocena, jak pokazuje badanie, może być z pełnym przekonaniem błędna. Uczestnicy nie byli nowicjuszami zmagającymi się z nową zabawką. Byli stałymi współtwórcami dużych, dojrzałych projektów open source, które dobrze znali, i mieli już pewne doświadczenie z narzędziami.

Co zrobili autorzy

- Przeprowadzili **randomizowane badanie kontrolowane** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Szesnastu doświadczonych programistów open source pracowało w dużych repozytoriach, do których regularnie wносиło wkład i które dobrze znało — średnio od pięciu lat w konkretnych projektach.
- Wykorzystali **246 rzeczywistych zadań** — poprawek błędów, nowych funkcji i refaktoryzacji pochodzących z systemów zgłoszeń samych projektów. Każde zadanie losowo przydzielono do warunku „AI dozwolona” albo „AI niedozwolona”.
- „AI dozwolona” oznaczała **narzędzia z początku 2025 roku: Cursor Pro z Claude 3.5/3.7 Sonnet**. Głównym pomiarem był rzeczywisty czas ukończenia każdego zadania. Zespół zebrał też prognozy programistów sprzed pracy i ich oszacowania po niej oraz przewidywania ekspertów ekonomii i uczenia maszynowego.

Co odkryli

- **Z AI zadania trwały o 19% dłużej.** Nie krócej — dłużej. 95-procentowy przedział ufności rozciąga się od około +2% do +39%, więc kierunek wyniku jest solidny, nawet jeśli dokładna wielkość mniej pewna.
- **Wszyscy przewidywali coś przeciwnego.** Programiści prognozowali przyspieszenie o 24%, eksperci uczenia maszynowego o około 38%, a ekonomiści o około 39%. Wszystkie trzy grupy oczekiwały dużej oszczędności czasu; stoper wykazał jego dodatkowy koszt.
- **Luka w postrzeganiu.** Choć po pracy z AI okazali się wolniejsi, programiści nadal oceniali, że przyspieszyła ich o około 20% — między odczuciem a pomiarem powstała różnica około 40 punktów procentowych.
- **Możliwe przyczyny, rozważone, a nie udowodnione.** Autorzy wskazują czynniki mogące wyjaśnić spowolnienie: programiści bardzo dobrze znają własne bazy kodu, więc asystent ma mniej do dodania; dojrzałe projekty mają wysokie, często niejawne standardy jakości; repozytoria są duże i pełne kontekstu niedostępnego modelowi; rzeczywisty czas pochłaniania pisanie poleceń, a następnie sprawdzanie i poprawianie wyników AI. Przedstawiają to jako tropy, nie rozstrzygnięcia.

Czego to nie pokazuje

- Badanie **nie** pokazuje, że AI nie przyspiesza większości programistów. Autorzy mówią to wprost: szesnastu ekspertów pracujących nad kodem znanym na pamięć nie reprezentuje przeciętnego programisty wykonującego przeciętne zadanie.
- **Nie** pokazuje, że AI jest bezużyteczna ani że spowalnia w innych warunkach — nowych osób w bazie kodu, projektach budowanych od zera, nieznanymi językami czy zupełnie innych dziedzinach.

- **Nie** zamraża narzędzi w czasie. To Cursor i Claude 3.5/3.7 Sonnet z początku 2025 roku; autorzy wyraźnie zaznaczają, że lepsze narzędzia lub lepsze sposoby użycia tych samych mogą zmienić wynik nawet w identycznych warunkach.
- Jest to **preprint** (opublikowany w lipcu 2025 roku, jeszcze bez recenzji naukowej), a autorzy przyznają, że nie mogą całkowicie wykluczyć artefaktów eksperymentalnych — choć wynik utrzymywał się w różnych analizach.
- Badanie **nie** pozwala na pocieszającą interpretację, że programiści musieli zyskać w inny sposób — nauczyć się więcej, czuć się lepiej albo pisać lepszy kod. Jedna zmierzona tutaj rzecz, odczuwane przyspieszenie, jest dokładnie tym, czemu przeczą dane.

Jak silne są dowody

- **Projekt badania jest wyjątkowo uczciwy.** Randomizacja, prawdziwe zadania, prawdziwe repozytoria i rzeczywisty pomiar czasu to duży krok naprzód wobec samoocen i rankingów benchmarków, na których opiera się większość twierdzeń o programowaniu z AI. Spowolnienie o 19% przetrwało testy odporności autorów.
- **Najbardziej przenośnym wynikiem jest luka w postrzeganiu.** Ekspercka intuicja na temat własnego przyspieszenia z AI myliła się o około 40 punktów procentowych, w optymistycznym kierunku. To przestroga dotycząca *każdego* deklarowanego wzrostu produktywności dzięki AI — włącznie z liczbami tego badania.
- **To migawka, nie trend.** Własne badanie uzupełniające METR z lutego 2026 roku, obejmujące podobnych programistów i nowsze narzędzia, wskazuje na przyspieszenie — bardzo z grubsza 18% u powracających programistów i 4% u nowych — ale autorzy oznaczają je jako słaby dowód, zniekształcony doborem chętnych (programiści coraz częściej odmawiali pracy bez AI, stawka spadła, a wybór zadań się przesunął). Najuczciwiej odczytać to tak: obraz się zmienia, a nawet tę zmianę autorzy raportują bez podkręcania wyniku.

Dlaczego to ważne

Większość sporu o AI i programowanie opiera się na demonstracjach i odczuciach: efektownym nagraniu ekranu, pewnym sobie twierdzeniu albo lekceważącym przewróceniu oczami. Rzadkie i warte uwagi jest to, że ktoś przeprowadził żmudny eksperyment — zastosował randomizację, zmierzył czas prawdziwej pracy, a potem zapytał ludzi o wrażenia. Odpowiedź jest niewygodna dla obu obozów. Podważa opowieść, że AI zawsze gwałtownie przyspiesza ekspertów pracujących nad trudnym, dobrze znanym kodem. Podważa też schludne przeciwieństwo — „udowodniono, że AI spowalnia programistów” — ponieważ nowsze dane tego samego zespołu wskazują już w drugą stronę. Najtrwalsza lekcja jest zarazem najmniejsza i najbardziej ludzka: osoby wykonujące pracę czuły się szybsze, choć pomiar wykazał spowolnienie. „Wydaje się szybsze” nie jest dowodem, że takie jest. Trzeba mierzyć.

Zwięzłe podsumowanie

W randomizowanym badaniu kontrolowanym METR zlecił szesnastu doświadczonym programistom open source wykonanie 246 rzeczywistych zadań w dobrze znanych im bazach kodu, losowo zezwalając na użycie narzędzi AI z początku 2025 roku (Cursor Pro oraz Claude 3.5/3.7 Sonnet) w połowie zadań. Programiści oczekiwali, że AI skróci czas o około 24%; zamiast tego wydłużyła go o 19% — a później nadal sądzili, że przyspieszyła ich o około 20%. Ta luka w postrzeganiu jest wyraźnym i odpornym rdzeniem badania. Dotyczy ono jednak szesnastu programistów, jednego wąskiego środowiska i określonego momentu na początku 2025 roku; autorzy podkreślają, że nie pokazuje, iż AI nie pomaga większości programistów, a ich własne badanie uzupełniające z 2026 roku już wskazuje na przyspieszenie, choć ma własne zastrzeżenia. Staranny pomiar, który warto traktować poważnie — nie werdykt o programowaniu z AI.

Źródła

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/>