The Clean Paper

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

Досвідченим розробникам здавалося, що з ШІ вони працюють швидше, хоча вимірювання показали протилежне — ось справжній результат, а не вирок ШІ-програмуванню

У рандомізованому контрольованому дослідженні METR шістнадцять досвідчених розробників відкритого ПЗ виконували 246 реальних завдань у добре знайомих їм кодових базах; для випадкової половини завдань були дозволені інструменти ШІ початку 2025 року (Cursor Pro разом із Claude 3.5/3.7 Sonnet). Вони очікували, що ШІ скоротить час виконання приблизно на 24%; натомість він збільшив його на 19%, а після роботи розробники все одно вважали, що ШІ прискорив їх приблизно на 20%. Саме цей розрив у сприйнятті — найчіткіше й найстійкіше ядро дослідження. Але це шістнадцять розробників, одне вузьке середовище та фіксований знімок початку 2025 року: автори прямо кажуть, що результат не показує, ніби ШІ не допомагає більшості розробників, а їхнє власне продовження 2026 року вже схиляється до прискорення — зі своїми застереженнями.

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

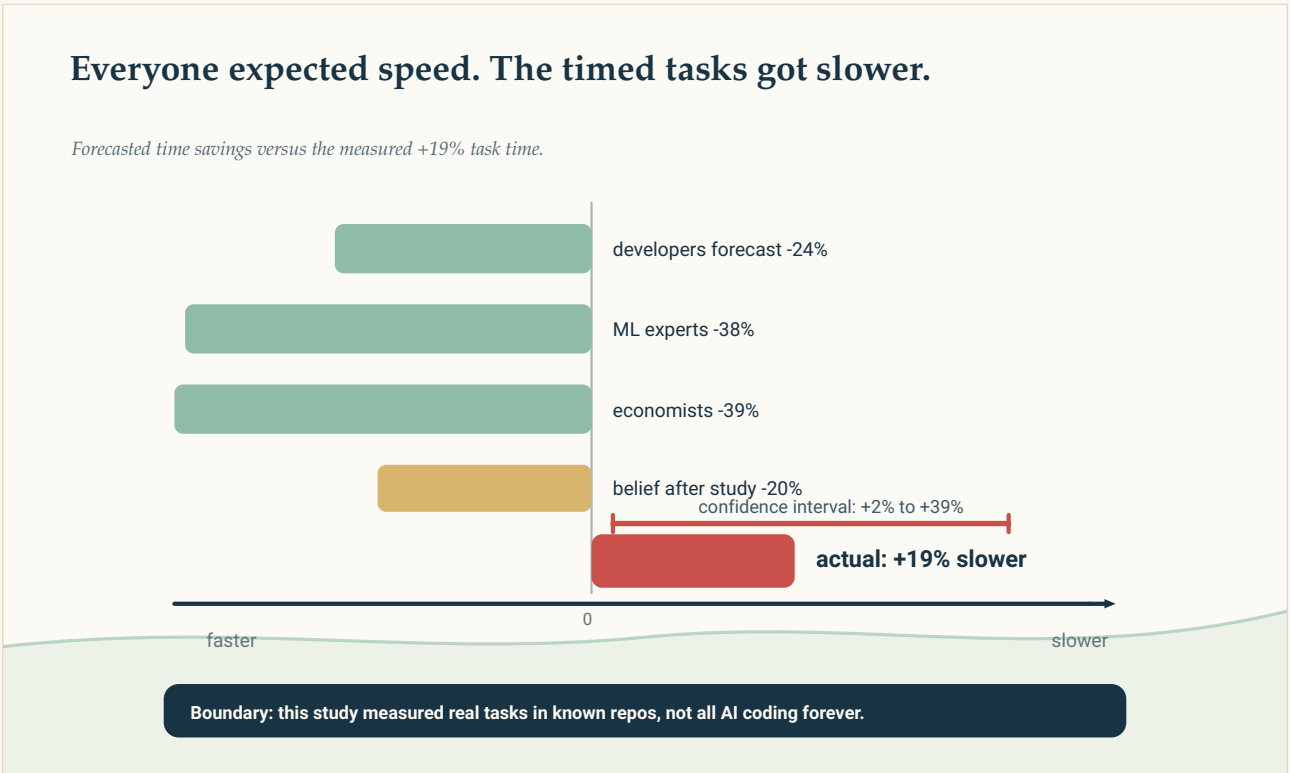
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Досвідченим розробникам здавалося, що з ШІ вони працюють швидше, хоча вимірювання показали протилежне — ось справжній результат, а не вирок ШІ-програмуванню

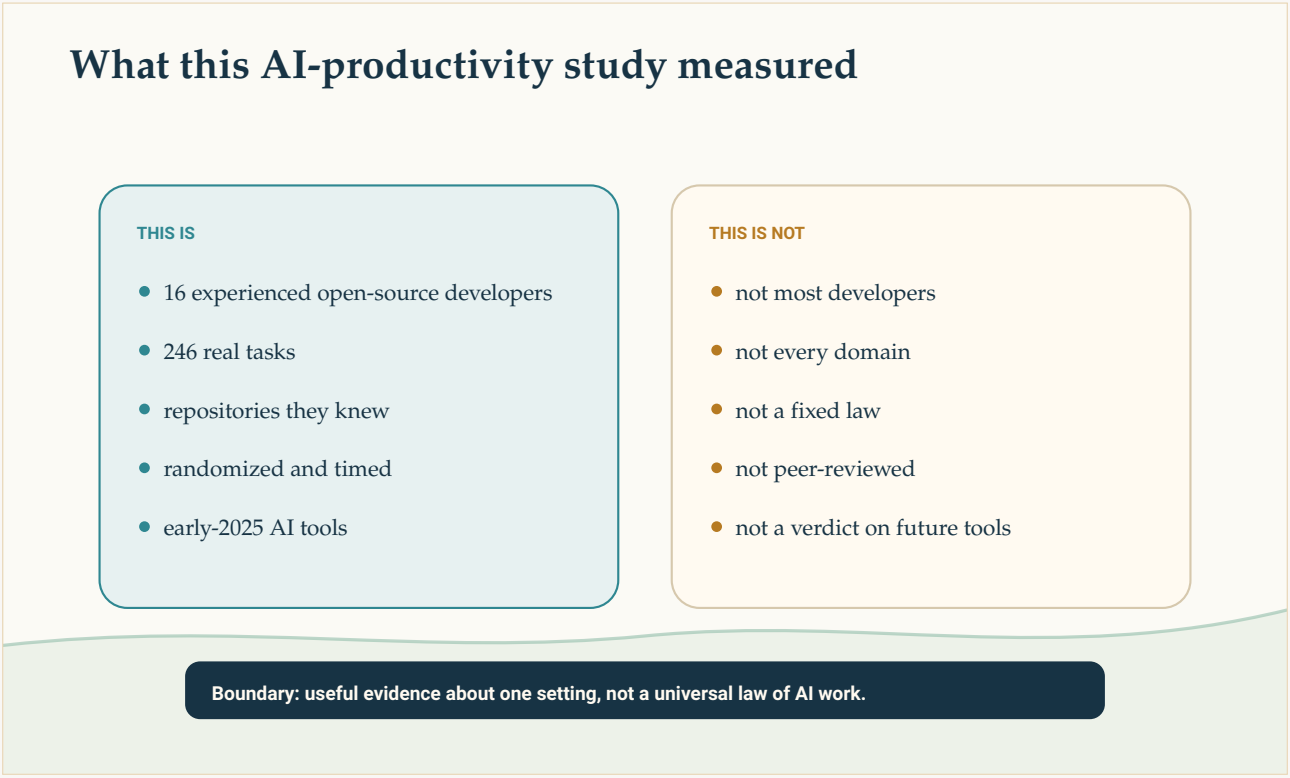
Запитайте досвідченого програміста, чи прискорює його роботу асистент для програмування на основі ШІ, — і зазвичай почуєте конкретне число: економить мені двадцять, тридцять відсотків часу. Запитайте економіста або дослідника машинного навчання — і число стане ще більшим. На початку 2025 року команда METR зробила повільну й дорогу річ: перевірила це. Дослідники залучили шістнадцять досвідчених розробників відкритого ПЗ, дали їм 246 реальних завдань із великих кодових баз, які ті добре знали, і — підкидаючи монету для кожного завдання — або дозволяли використовувати інструменти ШІ, або ні. А потім засікали час.

Розробники прогнозували, що ШІ скоротить час виконання завдань приблизно на 24%. Сталося навпаки: завдання, виконані з ШІ, потребували **на 19% більше часу**. І ось частина, над якою варто затриматися: навіть після завершення роботи ті самі розробники все ще вважали, що ШІ прискорив їх приблизно на 20%. Вони працювали повільніше, але відчували, що швидше, — і саме відстань між цими двома цифрами є найцікавішою частиною дослідження.

Це реальний, ретельно виміряний результат. Але це також лише шістнадцять розробників, які працювали з репозиторіями, відомими їм до дрібниць, і користувалися інструментами початку 2025 року. Тож висновок тут не зводиться до плаского «ШІ уповільнює розробників». Пізніші дані тієї самої команди вже вказують у протилежний бік.



Усі прогнозували, що ШІ прискорить роботу: розробники — на 24%, фахівці з машинного навчання — на 38%, економісти — на 39%, а самі розробники після експерименту вважали, що стали швидшими на 20%. Секундомір показав протилежне: +19% до часу виконання, з інтервалом від +2% до +39%. Розрив між відчуттям і вимірюванням — і є ключовим результатом.



Що саме вимірювало дослідження: 16 експертних розробників відкритого ПЗ, 246 реальних завдань, знайомі репозиторії, інструменти початку 2025 року та рандомізація. І чого воно не вимірювало: не більшість розробників, не інші галузі, не незмінний закон; дослідження також ще не пройшло рецензування.

Що саме тут вимірювали і навіщо потрібне рандомізоване дослідження

Рандомізоване контрольоване дослідження (RCT) — це інструмент, яким медицина відрізняє справжній ефект від бажаного. Тут кожне з 246 завдань випадково призначали до групи, де ШІ був дозволений, або до групи, де він був заборонений. Отже, в середньому єдина систематична різниця між двома наборами завдань — саме наявність ШІ. Завдяки цьому можна сказати, що ШІ *спричинив* зміну часу, а не просто помітити, що люди, які тягнуться до ШІ, з якихось інших причин і так працюють швидше чи повільніше. Це важливо, бо звичні докази приросту продуктивності від ШІ в програмуванні — самооцінки та результати бенчмарків — такого не дозволяють: бенчмарк не є реальною роботою, а самооцінка, як показує це дослідження, може бути впевнено хибною.

> Розробники тут не були новачками, які незграбно освоюють нову іграшку. Це були досвідчені учасники великих зрілих проєктів відкритого ПЗ, які добре знали свої кодові бази й уже мали певний досвід роботи з такими інструментами.

Що зробили автори

- Провели **рандомізоване контрольоване дослідження** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Шістнадцять досвідчених розробників відкритого ПЗ працювали кожен із великим репозиторієм, до якого регулярно робили внески й який добре знали; у середньому вони працювали над конкретними проєктами п’ять років.
- Використали **246 реальних завдань** — виправлення помилок, нові функції та рефакторинг із власних трекерів завдань цих проєктів. Кожне завдання випадково потрапляло до умов «ШІ дозволено» або «ШІ заборонено».
- «ШІ дозволено» означало **інструменти початку 2025 року: Cursor Pro з Claude 3.5/3.7 Sonnet.** Основним показником був реальний час виконання кожного завдання. Крім нього, команда зібрала прогнози розробників до роботи та їхні оцінки після неї, а також прогнози експертів з економіки й машинного навчання.

Що вони виявили

- **Із ШІ завдання виконувалися на 19% довше.** Не швидше, а повільніше. 95% довірчий інтервал простягається приблизно від +2% до +39%, тож напрямок ефекту досить надійний, навіть якщо точна його величина менш певна.
- **Усі передбачали протилежне.** Розробники прогнозували прискорення на 24%, фахівці з машинного навчання — приблизно на 38%, економісти — приблизно на 39%. Усі три групи очікували суттєвої економії часу; секундомір показав певну його втрату.
- **Розрив у сприйнятті.** Після того як вони виконали роботу повільніше, розробники все одно оцінювали, що ШІ прискорив їх приблизно на 20%. Це розрив близько 40 процентних пунктів між відчуттям і тим, що зафіксував годинник.
- **Можливі причини — зважені, але не доведені.** Автори перелічують чинники, які могли спричинити уповільнення: ці розробники дуже добре знають власні кодові бази, тому асистентові є менше що додати; зрілі проєкти мають високі й часто неявні стандарти якості; репозиторії великі й містять багато контексту, якого модель не бачить; реальний час витрачається на формулювання запитів, а потім на перевірку й виправлення результатів ШІ. Автори подають це як можливі пояснення, а не як встановлені причини.

Чого це не показує

- Це **не** показує, що ШІ не здатний прискорювати більшість розробників. Автори кажуть про це прямо: шістнадцять експертів, які знають свій код майже напам'ять, — не те саме, що середній розробник із середнім завданням.
- Це **не** показує, що ШІ марний або що він уповільнює людей в інших умовах — наприклад, новачків у кодовій базі, роботу над новим проєктом, незнайомі мови програмування чи зовсім інші сфери.
- Воно **не** фіксує інструменти назавжди в одному стані. Тут ідеться про Cursor та Claude 3.5/3.7 Sonnet початку 2025 року; автори прямо зазначають, що кращі інструменти або кращі способи використання цих самих інструментів можуть змінити результат навіть у точно такому самому середовищі.
- Це **препринт** (опублікований у липні 2025 року, ще не рецензований), і автори визнають, що не можуть повністю виключити експериментальні артефакти, хоча результат зберігався в різних варіантах їхнього аналізу.
- Воно **не** дає підстав для заспокійливого пояснення, ніби розробники неодмінно виграли в чомусь іншому — більше навчилися, були задоволеніші чи написали якісніший код. Єдина річ такого типу, яку тут вимірювали, — відчуте прискорення — якраз суперечить даним.

Наскільки сильні докази

- **Дизайн незвично чесний.** Рандомізація, реальні завдання, реальні репозиторії та фактичне вимірювання часу — великий крок уперед порівняно із самооцінками й таблицями лідерів бенчмарків, на яких ґрунтується більшість тверджень про ШІ-програмування. Уповільнення на 19% зберігалося в перевірках стійкості результату, проведених авторами.
- **Найбільш переносний результат — розрив у сприйнятті.** Експертна інтуїція щодо власного прискорення завдяки ШІ помилилася приблизно на 40 процентних пунктів — у оптимістичний бік. Це застереження щодо *будь-яких* самозвітів про приріст продуктивності від ШІ, включно з числами з цього самого дослідження.
- **Це знімок моменту, а не тренд.** Власне продовження METR у лютому 2026 року, з подібними розробниками, але новішими інструментами, вже вказує на прискорення — дуже приблизно −18% часу для розробників, які повернулися до дослідження, і −4% для нових учасників. Проте автори називають ці дані слабкими: на них вплинуло те, хто погодився брати участь (розробники дедалі частіше відмовлялися працювати без ШІ, оплата знизилася, а вибір завдань змістився). Обережне прочитання таке: картина рухається, і навіть цей рух автори описують без спроб підштовхнути ваги в потрібний бік.

Чому це важливо

Більшість суперечок про ШІ та програмування живе на демонстраціях і відчуттях: ефектний запис екрана, упевнене твердження, у відповідь — скептичне закочування очей. Рідкість тут у тому, що хтось провів нудний експеримент: випадково розподілив умови, виміряв час реальної роботи, а потім запитав людей, як, на їхню думку, усе минуло. Відповідь незручна для обох таборів. Вона пробиває діру в історії про те, що ШІ однаково сильно турбоприскорює досвідчених розробників на складному й добре знайомому коді. І так само пробиває діру в акуратній протилежності — «доведено, що ШІ уповільнює розробників», — бо новіші дані тієї самої команди вже схиляються в інший бік. Найстійкіший урок водночас найменший і найлюдяніший: люди, які виконували роботу, відчували себе швидшими, коли об'єктивно працювали повільніше. «Так здається швидше» — не доказ того, що це справді швидше. Треба вимірювати.

Короткий підсумок

У рандомізованому контрольованому дослідженні METR шістнадцять досвідчених розробників відкритого ПЗ виконали 246 реальних завдань у добре знайомих їм кодових базах; для випадкової половини завдань були дозволені інструменти ШІ початку 2025 року — Cursor Pro разом із Claude 3.5/3.7 Sonnet. Розробники очікували, що ШІ скоротить час виконання приблизно на 24%; натомість він збільшив його на 19%, а після роботи вони все одно вважали, що ШІ прискорив їх приблизно на 20%. Саме цей розрив у сприйнятті — найчіткіше й найстійкіше ядро дослідження. Але це шістнадцять розробників, одне вузьке середовище та фіксований знімок початку 2025 року; автори прямо кажуть, що результат не показує, ніби ШІ не допомагає більшості розробників, а їхнє власне продовження 2026 року вже вказує на прискорення — зі своїми застереженнями. Ретельне вимірювання, яке варто сприймати серйозно, але не вирок ШІ-програмуванню.

Джерела

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/