



Опытным разработчикам казалось, что с ИИ они работают быстрее, хотя измерения показали обратное — вот настоящий результат, а не приговор ИИ-программированию

В рандомизированном контролируемом исследовании METR шестнадцать опытных разработчиков открытого ПО выполнили 246 реальных задач в хорошо знакомых им кодовых базах; для случайной половины задач разрешались ИИ-инструменты начала 2025 года (Cursor Pro вместе с Claude 3.5/3.7 Sonnet). Они ожидали, что ИИ сократит время выполнения примерно на 24%; вместо этого он увеличил его на 19%, а после работы разработчики всё равно считали, что ИИ ускорил их примерно на 20%. Именно этот разрыв в восприятии — самое ясное и устойчивое ядро исследования. Но это шестнадцать разработчиков, одна узкая среда и фиксированный снимок начала 2025 года: авторы прямо говорят, что результат не показывает, будто ИИ не помогает большинству разработчиков, а их собственное продолжение 2026 года уже склоняется в сторону ускорения — со своими оговорками.

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

Опытным разработчикам казалось, что с ИИ они работают быстрее, хотя измерения показали обратное — вот настоящий результат, а не приговор ИИ-программированию

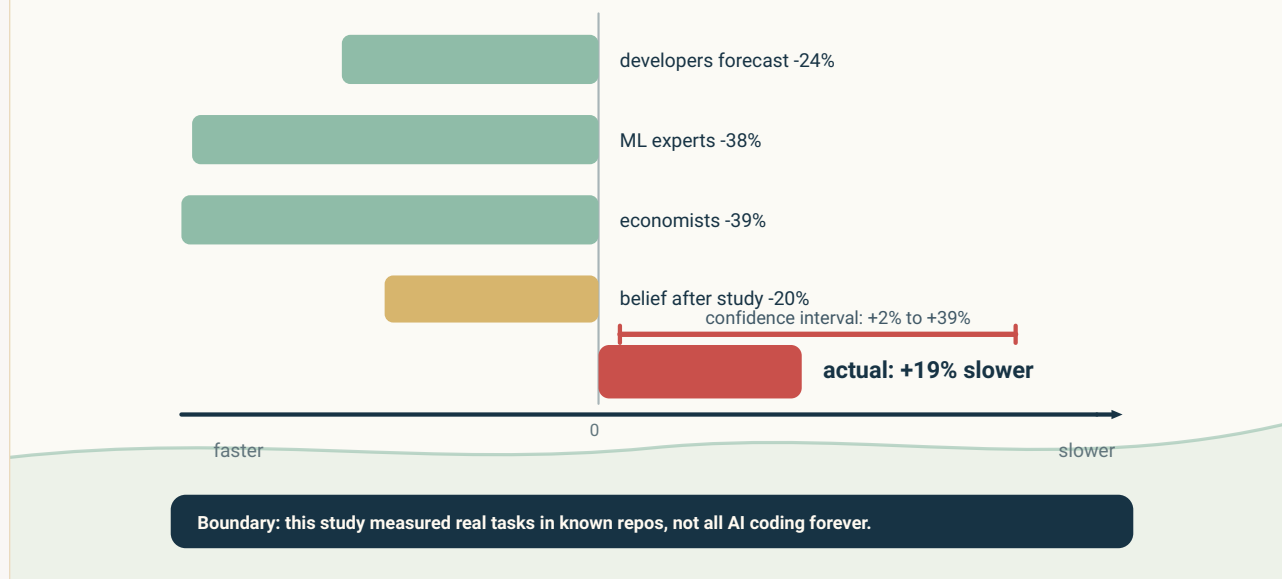
Спросите опытного программиста, ускоряет ли его работу ИИ-ассистент для написания кода, — и обычно услышите конкретную цифру: экономит мне двадцать, тридцать процентов времени. Спросите экономиста или исследователя машинного обучения — и цифра станет ещё больше. В начале 2025 года команда METR сделала медленную и дорогую вещь: проверила это. Исследователи набрали шестнадцать опытных разработчиков открытого ПО, дали им 246 реальных задач из крупных кодовых баз, которые те хорошо знали, и — случайным образом для каждой задачи — либо разрешали использовать ИИ-инструменты, либо нет. А затем засекали время.

Разработчики прогнозировали, что ИИ сократит время выполнения задач примерно на 24%. Произошло обратное: задачи, выполнявшиеся с ИИ, требовали **на 19% больше времени**. И вот на чём стоит задержаться: даже закончив работу, те же разработчики по-прежнему считали, что ИИ ускорил их примерно на 20%. Они работали медленнее, но ощущали, что быстрее, — и именно расстояние между этими двумя цифрами является самой интересной частью исследования.

Это реальный и тщательно измеренный результат. Но это также всего шестнадцать разработчиков, работавших с репозиториями, которые они знали до мелочей, и использовавших инструменты начала 2025 года. Поэтому вывод нельзя свести к плоскому «ИИ замедляет разработчиков». Более поздние данные той же команды уже указывают в противоположную сторону.

Everyone expected speed. The timed tasks got slower.

Forecasted time savings versus the measured +19% task time.



Все ожидали, что ИИ ускорит работу: разработчики — на 24%, специалисты по машинному обучению — на 38%, экономисты — на 39%, а сами разработчики после эксперимента считали, что стали быстрее на 20%. Секундомер показал обратное: +19% ко времени выполнения, с интервалом от +2% до +39%. Разрыв между ощущением и измерением — и есть ключевой результат.

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.

Что именно измеряло исследование: 16 опытных разработчиков открытого ПО, 246 реальных задач, знакомые репозитории, инструменты начала 2025 года и рандомизация. И чего оно не измеряло: не большинство разработчиков, не другие отрасли, не неизменный закон; исследование также ещё не прошло рецензирование.

Original diagram — The Clean Paper CC BY 4.0

Что именно здесь измеряли и зачем нужно рандомизированное исследование

Рандомизированное контролируемое исследование (RCT) — инструмент, которым, например, медицина отделяет реальный эффект от желаемого. Здесь каждую из 246 задач случайно относили либо к условию, где ИИ разрешён, либо к условию, где он запрещён. Поэтому в среднем единственное систематическое различие между двумя наборами задач — именно наличие ИИ. Это позволяет сказать, что ИИ *вызвал* изменение времени, а не просто заметить, что люди, тянущиеся к ИИ, по каким-то другим причинам и так работают быстрее или медленнее. Это важно, потому что обычные доказательства прироста продуктивности от ИИ в программировании — самооценки и результаты бенчмарков — такого вывода не позволяют: бенчмарк не является реальной работой, а самооценка, как показывает это исследование, может быть уверенно ошибочной. Разработчики здесь не были новичками, неловко осваивающими новую игрушку. Это были опытные участники крупных зрелых проектов открытого ПО, хорошо знавшие свои кодовые базы и уже имевшие некоторый опыт работы с такими инструментами.

Что сделали авторы

- Провели **рандомизированное контролируемое исследование** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein). Шестнадцать опытных разработчиков открытого ПО работали каждый с крупным репозиторием, в который регулярно вносили вклад и который хорошо знали; в среднем они работали над конкретными проектами пять лет.
- Использовали **246 реальных задач** — исправления ошибок, новые функции и рефакторинг из собственных трекеров задач этих проектов. Каждая задача случайно попадала в условие «ИИ разрешён» или «ИИ запрещён».
- «ИИ разрешён» означало **инструменты начала 2025 года: Cursor Pro с Claude 3.5/3.7 Sonnet**. Основным показателем было реальное время выполнения каждой задачи. Кроме него команда собрала прогнозы разработчиков до работы и их оценки после неё, а также прогнозы экспертов по экономике и машинному обучению.

Что они обнаружили

- **С ИИ задачи выполнялись на 19% дольше.** Не быстрее, а медленнее. 95%-й доверительный интервал простирается примерно от +2% до +39%, поэтому направление эффекта достаточно надёжно, даже если его точная величина менее определённа.
- **Все предсказали обратное.** Разработчики прогнозировали ускорение на 24%, специалисты по машинному обучению — примерно на 38%, экономисты — примерно на 39%. Все три группы ожидали существенной экономии времени; секундомер показал некоторую его потерю.
- **Разрыв в восприятии.** После того как разработчики выполнили работу медленнее, они всё равно оценивали, что ИИ ускорил их примерно на 20%. Это разрыв примерно в 40 процентных пунктов между ощущением и тем, что зафиксировали часы.
- **Возможные причины взвешены, но не доказаны.** Авторы перечисляют факторы, которые могли вызвать замедление: эти разработчики очень хорошо знают собственные кодовые базы, поэтому ассистенту остаётся меньше что добавить; зрелые проекты имеют высокие и часто неявные стандарты качества; репозитории велики и содержат много контекста, которого модель не видит; реальное время уходит на формулирование запросов, а затем на проверку и исправление результатов ИИ. Авторы представляют это как возможные объяснения, а не как установленные причины.

Чего это не показывает

- Это **не** показывает, что ИИ не способен ускорять большинство разработчиков. Авторы говорят об этом прямо: шестнадцать экспертов, знающих свой код почти наизусть, — не то же самое, что средний разработчик со средней задачей.
- Это **не** показывает, что ИИ бесполезен или что он замедляет людей в других условиях — например, новичков в кодовой базе, работу над новым проектом, незнакомые языки программирования или совсем другие области.
- Оно **не** фиксирует инструменты навсегда в одном состоянии. Здесь речь идёт о Cursor и Claude 3.5/3.7 Sonnet начала 2025 года; авторы прямо отмечают, что лучшие инструменты или лучшие способы использования тех же инструментов могут изменить результат даже в точно такой же среде.
- Это **препринт** (опубликован в июле 2025 года, ещё не рецензирован), и авторы признают, что не могут полностью исключить экспериментальные артефакты, хотя результат сохранялся в различных вариантах их анализа.
- Оно **не** даёт оснований для утешительного объяснения, будто разработчики обязательно выиграли в чём-то другом — больше научились, были довольнее или написали более качественный код. Единственная вещь такого рода, которую здесь измеряли, — ощущаемое ускорение — как раз противоречит данным.

Насколько сильны доказательства

- **Дизайн необычно честный.** Рандомизация, реальные задачи, реальные репозитории и фактическое измерение времени — большой шаг вперёд по сравнению с самооценками и таблицами лидеров бенчмарков, на которых основано большинство заявлений об ИИ-программировании. Замедление на 19% сохранялось в проверках устойчивости результата, проведённых авторами.
- **Самый переносимый результат — разрыв в восприятии.** Экспертная интуиция о собственном ускорении благодаря ИИ ошиблась примерно на 40 процентных пунктов — в оптимистичную сторону. Это предупреждение относительно *любых* самоотчётов о приросте продуктивности от ИИ, включая цифры из этого же исследования.
- **Это снимок момента, а не тренд.** Собственное продолжение METR в феврале 2026 года, с похожими разработчиками, но более новыми инструментами, уже указывает на ускорение — очень приблизительно — 18% времени для разработчиков, вернувшихся в исследование, и —4% для новых участников. Однако авторы называют эти данные слабыми: на них повлияло то, кто соглашался участвовать (разработчики всё чаще отказывались работать без ИИ, оплата снизилась, а выбор задач сместился). Осторожное прочтение таково: картина меняется, и даже это движение авторы описывают без попытки наклонить весы в нужную сторону.

Почему это важно

Большинство споров об ИИ и программировании живёт на демонстрациях и ощущениях: эффектная запись экрана, уверенное заявление, в ответ — скептическое закатывание глаз. Редкость здесь в том, что кто-то провёл скучный эксперимент: случайно распределил условия, измерил время реальной работы, а затем спросил людей, как, по их мнению, всё прошло. Ответ неудобен для обоих лагерей. Он пробивает дыру в истории о том, что ИИ одинаково сильно турбоускоряет опытных разработчиков на сложном и хорошо знакомом коде. И точно так же пробивает дыру в аккуратной противоположности — «доказано, что ИИ замедляет разработчиков», — потому что более новые данные той же команды уже склоняются в другую сторону. Самый устойчивый урок одновременно самый маленький и самый человеческий: люди, выполнявшие работу, чувствовали себя быстрее, когда объективно работали медленнее. «Так кажется быстрее» — не доказательство того, что это действительно быстрее. Нужно измерять.

Краткий итог

В рандомизированном контролируемом исследовании METR шестнадцать опытных разработчиков открытого ПО выполнили 246 реальных задач в хорошо знакомых им

кодовых базах; для случайной половины задач были разрешены ИИ-инструменты начала 2025 года — Cursor Pro вместе с Claude 3.5/3.7 Sonnet. Разработчики ожидали, что ИИ сократит время выполнения примерно на 24%; вместо этого он увеличил его на 19%, а после работы они всё равно считали, что ИИ ускорил их примерно на 20%. Именно этот разрыв в восприятии — самое ясное и устойчивое ядро исследования. Но это шестнадцать разработчиков, одна узкая среда и фиксированный снимок начала 2025 года; авторы прямо говорят, что результат не показывает, будто ИИ не помогает большинству разработчиков, а их собственное продолжение 2026 года уже указывает на ускорение — со своими оговорками. Тщательное измерение, которое стоит воспринимать серьёзно, но не приговор ИИ-программированию.

Источники

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/>