



The CLEAN PAPER

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

有經驗的開發者在可測量的變慢中感覺更快—— 真實的發現，以及它不是的關於 AI 編程的判決

METR 的一項隨機對照試驗讓十六位經驗豐富的開源開發者在他們熟知的代碼庫上完成 246 個真實任務，隨機一半允許使用 2025 年初的 AI 工具（Cursor Pro + Claude 3.5/3.7 Sonnet）。他們預期 AI 削減任務時間約 24%；結果完成時間反而增加了 19%——事後他們仍然相信 AI 讓他們加快了約 20%。這個感知差距是研究最尖銳、最穩健的核心。但這是十六位開發者，一個狹窄的環境，以及一個固定的 2025 年初快照：作者明確表示它並未顯示 AI 未能幫助大多數開發者，他們自己 2026 年的後續研究已偏向加速，帶有其自身的保留。

Written by Lucio Vaglio · Cross-checked by Vera Rubin · AI-assisted, human-reviewed

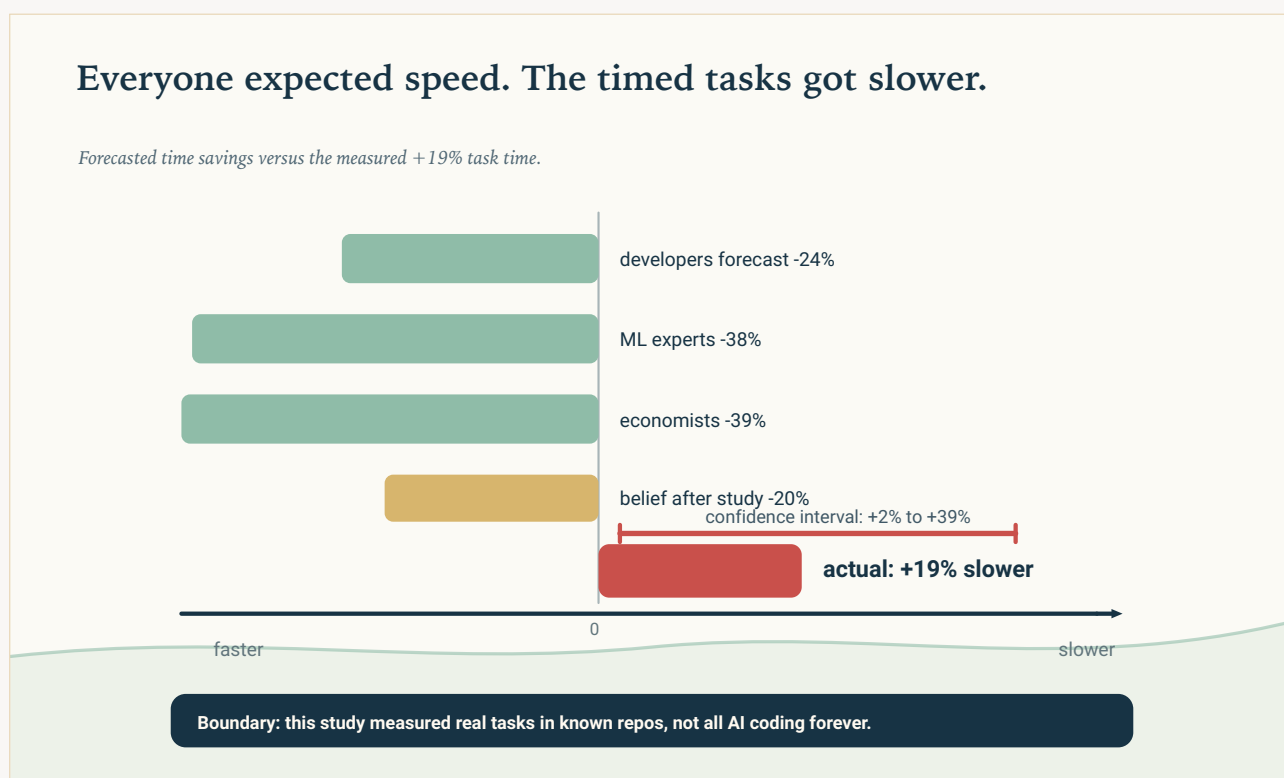
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

有經驗的開發者在可測量的變慢中感覺更快——真實的發現，以及它不是的關於 AI 編程的判決

問一個有經驗的程式設計師 AI 編程助手是否讓他們更快，你通常會得到一個數字：它為我節省了二十、三十個百分點。問一位經濟學家，或一位機器學習研究者，這個數字會更大。2025 年初，METR 的一個團隊做了那件緩慢、昂貴的事並進行了檢驗。他們找了十六位經驗豐富的開源開發者，交給他們 246 個來自他們熟悉的大型代碼庫的真實任務，然後——通過拋硬幣，逐個任務——要麼讓他們使用 AI 工具，要麼不讓他們使用。然後他們計時了工作。

開發者們預測 AI 能將他們的任務時間削減約 24%。它做了相反的事：用 AI 完成的任務用時**多出 19%**。而這裡是值得坐下來想想的部分——在完成後，同樣的開發者仍然相信 AI 讓他們加快了，大約 20%。他們變慢了，卻感覺變快了，這兩個數字之間的距離是這項研究中最有趣的東西。

這是一項真實的、仔細測量的結果。它也是十六位開發者，在他們極其熟悉的倉庫上，使用 2025 年初的工具——而且它不是那句平坦的句子「AI 讓開發者變慢」。同一團隊隨後的數據已經指向了相反的方向。



每個人都預測 AI 會加快工作速度——開發者 -24%，機器學習專家 -38%，經濟學家 -39%，以及開發者自己事後相信的 -20%。秒表發現 +19% 更慢，區間從 +2% 到 +39%。感覺與測量之間的差距才是發現。

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- not peer-reviewed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.

這項研究測量了什麼——16 位專家級開源開發者，246 個真實任務，熟悉的倉庫，2025 年初的工具，隨機化——以及它沒有測量什麼：不是大多數開發者，不是其他領域，不是一條固定法則，未經同行評議。

Original diagram —The Clean Paper CC BY 4.0

這測量了什麼，以及隨機化試驗為你買到了什麼

隨機對照試驗（RCT）是醫學用來區分真實效應和願望效應的工具。在這裡，246 個任務中的每一個都被隨機分配為允許或禁止使用 AI，因此平均而言，兩堆之間唯一的系統性差異就是 AI 本身。這就是讓你能夠說 AI 導致了時間變化，而不是僅僅注意到伸手使用 AI 的人恰好因為其他原因更快或更慢。這很重要，因為 AI 編程收益的通常證據——自我報告和基準分數——不能做到這一點：一個基準不是真實的工作，而一份自我報告，正如這項研究所顯示的，可以自信地是錯的。這裡的開發者不是在新玩具上摸索的新手。他們是他們熟知的大型、成熟開源項目的確定貢獻者，在使用這些工具方面有一些先前的經驗。

作者做了什麼

- 運行了一項**隨機對照試驗**（METR：Joel Becker、Nate Rush、Beth Barnes、David Rein）。十六位經驗豐富的開源開發者，每個人都在他們定期貢獻並熟知的大型倉庫上工作——在具體項目上平均有五年經驗。
- 使用了**246 個真實任務**——從這些項目自己的問題追蹤器中提取的錯誤修復、功能和重構。每個任務被隨機分配到「允許 AI」或「禁止 AI」。
- 「允許 AI」意味著**2025 年初的工具：Cursor Pro 加 Claude 3.5/3.7 Sonnet**。主要指標是每個任務的實際完成時間。同時，團隊收集了預測（從事前開發者那裡）和估算（從事後他們那裡），以及

來自經濟學和機器學習專家的預測。

他們發現了什麼

- **使用 AI 時，任務用時多出 19%。**不是更快——更慢。95% 置信區間大致從 +2% 到 +39%，因此方向是確定的，即使確切大小不是。
- **每個人都預測了相反的結果。**開發者預測了 24% 的加速；機器學習專家約 38%；經濟學家約 39%。三組人都預期 AI 能節省大量時間；秒表發現它花費了一些。
- **感知差距。**在完成工作並出來變慢之後，開發者仍然估計 AI 讓他們加快了約 20%——他們感覺到的和時鐘記錄之間的差距約為 40 個百分點。
- **候選原因，權衡而非證明。**作者排列了可以解釋減速的因素：這些開發者深度了解自己的代碼庫，因此助手能添加的較少；成熟項目帶有很高的、通常是隱含的質量標準；倉庫龐大且充滿模型不具備的上下文；而真實時間花在了提示上，然後審查和修正 AI 輸出。他們將這些作為線索呈現，而非判決。

這沒有顯示什麼

- 它**沒有**顯示 AI 未能加速大多數開發者。作者直接這樣說：十六位專家在他們瞭如指掌的代碼上不是普通開發者在普通任務上的樣子。
- 它**沒有**顯示 AI 是無用的，或者它在其他情境下讓人們變慢——代碼庫的新來者、綠地工作、不熟悉的語言，或完全不同的領域。
- 它**沒有**將工具凍結在時空中。這是 2025 年初的 Cursor 和 Claude 3.5/3.7 Sonnet；作者明確表示更好的工具，或使用這些工具的更好方式，即使在這種確切環境中也可能改變結果。
- 它是一篇**預印本**（2025 年 7 月發布，尚未經同行評議），作者指出他們不能完全排除實驗假象——儘管該發現在他們的分析中經受了檢驗。
- 它**沒有**授權那種令人安心的解讀，即開發者一定以某種其他方式獲得了——學到了更多、感覺更快樂、寫出了更好的代碼。這裡測量的唯一一件事，感覺上的加速，正是數據所否定的。

證據有多強

- **設計異常誠實。**隨機化，真實任務，真實倉庫，實際計時——比大多數關於 AI 編程的聲稱所依賴的自我報告和基準排行榜上升了一大步。19% 的減速經受了作者的穩健性檢查。
- **最具可移植性的發現是感知差距。**專家對自己 AI 加速的直覺錯了大約 40 個百分點，且偏向樂觀方向。這對每一份關於 AI 生產力提升的自我報告都是一個警示——包括這項研究自己的數字。
- 它**是一個快照，而非趨勢。**METR 自己在 2026 年 2 月的後續研究，在同樣類型的開發者上使用更新的工具，指向了加速——對於回訪開發者約 -18%，對於新開發者約 -4%——但作者將其標記為弱證據，被誰願意參與所扭曲（開發者越來越拒絕在沒有 AI 的情況下工作，報酬率下降，任務選擇偏斜）。誠實的解讀是圖景在移動，而即使是移動本身，在報告時也將拇指從秤上移開了。

為什麼重要

簡潔摘要

在一項隨機對照試驗中，METR 讓十六位經驗豐富的開源開發者在他們熟知的代碼庫上完成 246 個真實任務，隨機一半允許使用 2025 年初的 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。開發者預期 AI 削減任務時間約 24%；結果完成時間反而增加了 19%——事後他們仍然相信 AI 讓他們加快了約 20%。這個感知差距是研究最尖銳、最穩健的核心。但這是十六位開發者，一個狹窄的環境，以及一個固定的 2025 年初快照；作者明確表示它並未顯示 AI 未能幫助大多數開發者，他們自己 2026 年的後續研究已偏向加速，帶有其自身的保留。一項值得認真對待的仔細測量——不是對 AI 編程的判決。

關於 AI 和編程的爭論大多在演示和氛圍上運行：一段流暢的屏幕錄像，一個自信的主張，一個反駁的白眼。罕見的是，也是這篇文章值得一讀的，是有人跑了那個枯燥的實驗——隨機化，計時真實工作，然後問人們感覺如何。答案對兩個陣營都不舒服。它戳破了 AI 統一地在困難、熟悉的代碼上加速專家開發者的故事。它也戳破了整潔的反面——「AI 讓開發者變慢，已被證明」——因為同一團隊的新數據已經偏向另一邊。最持久的教訓也是最微小和最人性的：做這項工作的人感覺更快，而可測量地更慢。「感覺更快」不是它真正更快的證據。測量它。

來源

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/>