



The CLEAN PAPER

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

資深開發者覺得 AI 讓自己更快，實際卻量到更慢——真正的發現，以及它並不是什麼 AI 程式開發判決

METR 的一項隨機對照試驗，讓 16 名資深開源開發者在自己熟悉的程式碼庫中完成 246 個真實工作項目；每個任務隨機決定是否允許使用 2025 年初的 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。開發者原本預期 AI 能把完成時間縮短約 24%；結果反而增加了 19%。更有意思的是，做完後他們仍相信 AI 讓自己快了約 20%。這個「感覺與量測之間的落差」是研究最銳利、也最穩健的核心。但樣本只有 16 人、情境很窄，而且只是固定在 2025 年初工具狀態的一張快照；作者明確表示，這不能證明 AI 對多數開發者沒有幫助，而他們自己 2026 年的後續研究已經傾向顯示加速，雖然也有自己的限制。

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

資深開發者覺得 AI 讓自己更快，實際卻量到更慢——真正的發現，以及它並不是什麼 AI 程式開發判決

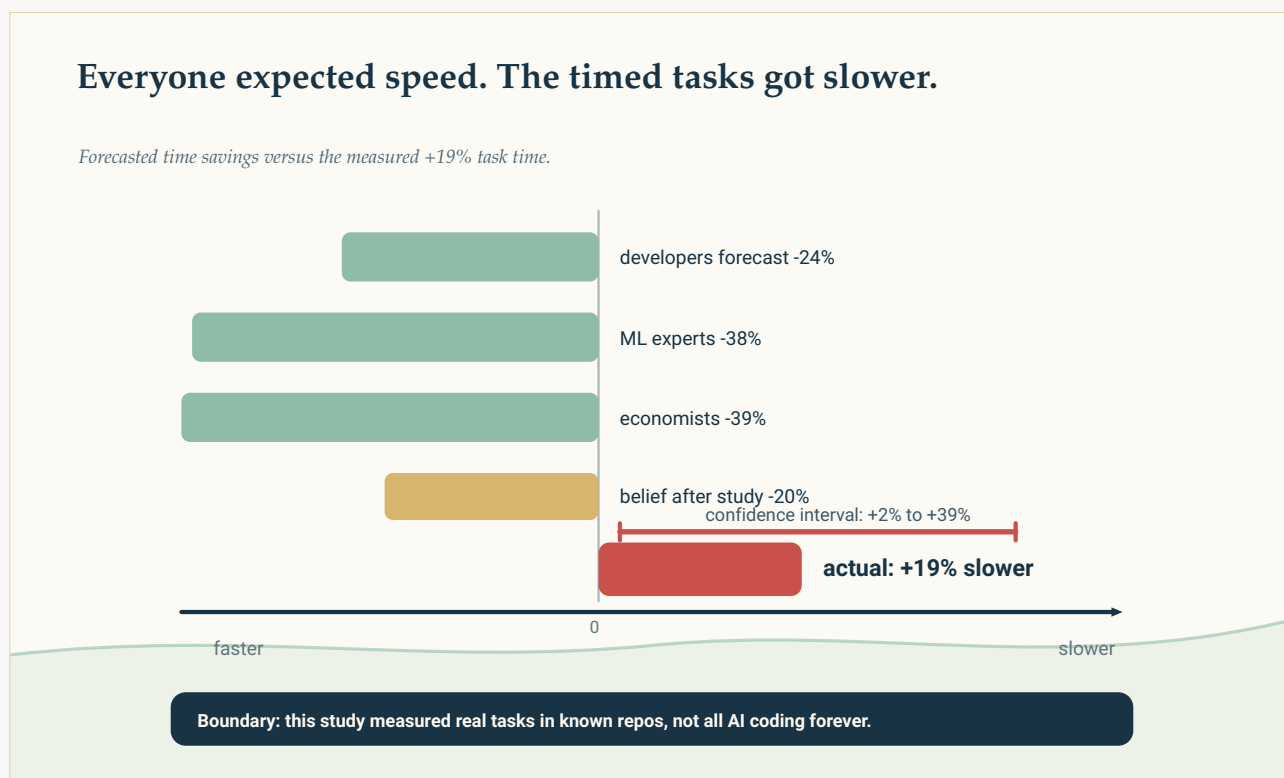
問一位有經驗的程式設計師：「AI 程式助理有沒有讓你更快？」通常很快就會得到一個數字：省了 20%、30% 的時間。問經濟學家或機器學習研究者，數字往往還會更大。

2025 年初，METR 的一個團隊做了一件緩慢、昂貴，但真正能回答問題的事：直接量。他們找來 16 名資深開源開發者，交給他們 246 個來自大型、而且本人非常熟悉的程式碼庫的真實任務；每一個任務都用隨機方式決定能不能使用 AI 工具。然後計時。

開發者原本預測，AI 會把任務時間縮短約 24%。結果正好相反：允許 AI 的任務反而**多花了 19% 時間**。

而更值得停下來看的，是後面這件事：做完實驗後，同一批開發者仍然相信 AI 讓自己快了約 20%。他們實際更慢，感覺卻更快；這兩個數字之間的距離，才是這項研究最有意思的部分。

這是一個真實、仔細量測的結果。但它同時也只是 16 名開發者、在自己非常熟悉的程式庫、使用 2025 年初工具的結果——而不是一句平面的「AI 讓開發者變慢」。同一個團隊較新的資料，甚至已經開始往相反方向走。



所有人都預測 AI 會加速工作：開發者 -24%、機器學習專家 -38%、經濟學家 -39%，連開發者完成實驗後的事後感受都還是 -20%。真正的計時器卻量到 +19%——變慢——信賴區間約 +2% 到 +39%。「感覺」與「量到」之間的落差，就是這項研究最值得帶走的結果。

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- not peer-reviewed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.

這項研究真正量到的是什麼：16 名資深開源開發者、246 個真實任務、熟悉的程式庫、2025 年初工具、隨機分派。它沒有量到什麼：不是多數開發者、不是其他領域、不是固定法則，也尚未同儕審查。

Original diagram —The Clean Paper CC BY 4.0

這項研究量了什麼，以及隨機試驗多給了你什麼

隨機對照試驗（randomized controlled trial，RCT）是醫學常用來區分「真正效果」與「看起來有希望」的工具。這裡 246 個任務中的每一個，都被隨機分派為可以或不可以用 AI；因此平均而言，兩堆任務唯一系統性的差別就是 AI 本身。

這讓你可以說，是 AI 造成了時間變化，而不是只觀察到「比較喜歡用 AI 的人剛好本來就比較快或比較慢」。這點非常重要，因為 AI 程式開發增益常見的證據——自我報告與基準測試分數——做不到這件事：基準測試不是真實工作，而自我報告，正如這項研究展示的，可以很有自信地錯。

這裡的開發者也不是第一次碰工具的新手。他們是大型、成熟開源專案的既有貢獻者，非常熟悉自己的程式庫，而且已有一定 AI 工具使用經驗。

作者做了什麼

- METR 的 Joel Becker、Nate Rush、Beth Barnes、David Rein 執行一項**隨機對照試驗**。16 名資深開源開發者，每人都在自己長期參與、非常熟悉的大型程式庫工作；平均已在對應專案工作約五年。
- 使用 **246 個真實任務**：來自那些專案 issue tracker 的 bug 修正、新功能與重構。每個任務隨機分成「可使用 AI」或「不可使用 AI」。

- 「可使用 AI」指的是 **2025 年初工具：Cursor Pro 搭配 Claude 3.5/3.7 Sonnet**。主要指標是每個任務真正花掉的完成時間。另外也收集開發者事前預測、事後估計，以及經濟學與機器學習專家的預測。

他們發現了什麼

- **使用 AI 時，任務反而多花 19% 時間**。不是更快，是更慢。95% 信賴區間大約從 +2% 到 +39%，因此方向相當穩固，精確效應大小則仍有不確定性。
- **所有人原本都猜錯方向**。開發者預測快 24%，機器學習專家約 38%，經濟學家約 39%。三群人都預期 AI 能省下大量時間；碼表卻顯示它多花了一些。
- **感知落差**。做完工作、實際變慢後，開發者仍估計 AI 讓自己快了約 20%——「感受到的」和「實際計時」之間，大約差 40 個百分點。
- **可能原因被列出、比較，但沒有被證明**。作者提出幾個可能解釋：這些開發者對自己程式庫太熟，助理能補充的東西較少；成熟專案有很高、而且常是默會的品質標準；程式庫很大，模型缺少許多脈絡；以及真實時間會消耗在提示、檢查與修正 AI 輸出。作者把這些當成線索，不是定論。

這並沒有證明什麼

- 它**沒有**證明 AI 無法加速大多數開發者。作者直接這樣說：16 名在自己熟悉程式碼上工作的專家，並不等於「平均開發者做平均任務」。
- 它**沒有**證明 AI 沒用，也沒有證明它在其他情境會讓人變慢——例如新加入某程式庫的人、全新專案、不熟悉的語言，或其他領域。
- 它**沒有**把工具永遠凍結在原地。這是 2025 年初的 Cursor 與 Claude 3.5/3.7 Sonnet；作者明確表示，工具變好，或同樣工具的使用方式變好，都可能讓即使完全相同的情境得到不同結果。
- 這是一篇**預印本**（2025 年 7 月發布，尚未同儕審查）；作者也表示無法完全排除實驗設計造成的假象——雖然 19% 變慢的結果經過多種穩健性分析仍存在。
- 它**也不允許**一個安慰式解讀：「也許開發者雖然慢一點，但學得更多、比較快樂，或程式碼更好。」這些可能性沒有在這裡被證明；研究真正量到的一件主觀東西——「覺得自己變快」——恰好是資料直接反駁的。

證據有多強

- **研究設計異常誠實**。隨機化、真實任務、真實程式庫、實際計時——比 AI 程式開發討論常依賴的自我報告與基準測試排行榜高出一大截。19% 變慢在作者的穩健性檢查中仍然存在。
- **最可能跨情境成立的結果，是感知落差**。專家對「AI 讓自己快多少」的直覺，往樂觀方向錯了約 40 個百分點。這對任何 AI 生產力自我報告都是警告——也包括這篇研究自己的受試者。
- **這是一張快照，不是一條趨勢線**。METR 自己在 2026 年 2 月的後續研究，用類似開發者與較新工具，已經開始指向加速——很粗略地說，回鍋開發者約 -18%，新加入者約 -4%。但作者自己把

它標成弱證據，因為參與者選擇偏誤變得很嚴重：越來越多開發者拒絕在不能用 AI 的條件下工作，報酬率下降，而且任務選擇也偏了。最誠實的讀法是：圖景正在移動，而且連這個移動本身都必須小心測。

為什麼重要

AI 與程式開發的大部分爭論，都靠 demo 和感覺運轉：一段流暢的螢幕錄影、一個很有信心的宣稱，再配上一個反方向的白眼。真正少見、也讓這篇值得看的，是有人真的去做了那個無聊的實驗：隨機分派、計時真實工作，再問大家感覺如何。

答案讓兩邊都不太舒服。它戳破了「AI 一定能讓熟悉複雜程式碼的專家全面暴增速度」的故事；也戳破了另一個整齊的相反故事——「已證明 AI 讓開發者變慢」——因為同一團隊的新資料已經開始往相反方向走。

最耐久的教訓反而最小、也最人性：做工作的人感覺更快，但量出來更慢。「感覺更快」不是它真的更快的證據。量它。

重點摘要

在一項隨機對照試驗中，METR 讓 16 名資深開源開發者完成 246 個來自熟悉程式庫的真實任務，其中一半隨機允許使用 2025 年初 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。開發者原本預期 AI 能把任務時間縮短約 24%；結果反而讓完成時間增加 19%，而做完後他們仍相信 AI 讓自己快了約 20%。這個感知落差，是研究最銳利也最穩健的核心。但樣本只有 16 人、情境很窄，而且只是 2025 年初工具狀態的一張快照；作者明確表示，這不能證明 AI 對多數開發者沒有幫助，而他們 2026 年的後續研究甚至已經傾向顯示加速，只是也有自己的限制。這是一個值得認真看待的仔細量測，不是對 AI 程式開發的最終判決。

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/>