



The CLEAN PAPER

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

有经验的开发者在可测量的变慢中感觉更快—— 真实的发现，以及它不是的关于 AI 编程的判决

METR 的一项随机对照试验让十六位经验丰富的开源开发者在他们熟知的代码库上完成 246 个真实任务，随机一半允许使用 2025 年初的 AI 工具（Cursor Pro + Claude 3.5/3.7 Sonnet）。他们预期 AI 削减任务时间约 24%；结果完成时间反而增加了 19%——事后他们仍然相信 AI 让他们加快了约 20%。这个感知差距是研究最尖锐、最稳健的核心。但这是十六位开发者，一个狭窄的环境，以及一个固定的 2025 年初快照：作者明确表示它并未显示 AI 未能帮助大多数开发者，他们自己 2026 年的后续研究已偏向加速，带有其自身的保留。

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

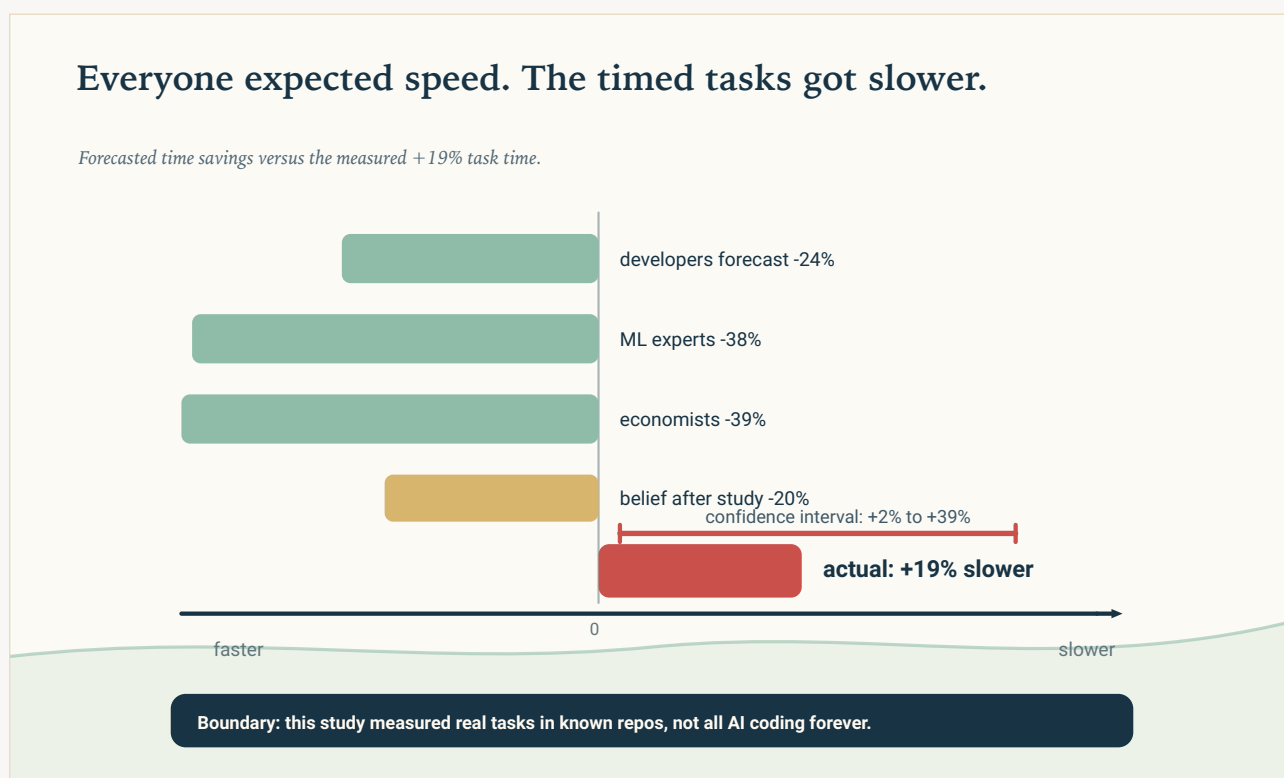
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

有经验的开发者在可测量的变慢中感觉更快——真实的发现，以及它不是的关于 AI 编程的判决

问一个有经验的程序员 AI 编程助手是否让他们更快，你通常会得到一个数字：它为我节省了二十、三十个百分点。问一位经济学家，或一位机器学习研究者，这个数字会更大。2025 年初，METR 的一个团队做了那件缓慢、昂贵的事并进行了检验。他们找了十六位经验丰富的开源开发者，交给他们 246 个来自他们熟悉的大型代码库的真实任务，然后——通过抛硬币，逐个任务——要么让他们使用 AI 工具，要么不让他们使用。然后他们计时了工作。

开发者们预测 AI 能将他们的任务时间削减约 24%。它做了相反的事：用 AI 完成的任务用时多出 19%。而这里是值得坐下来想想的部分——在完成后，同样的开发者仍然相信 AI 让他们加快了，大约 20%。他们变慢了，却感觉变快了，这两个数字之间的距离是这项研究中最有趣的东西。

这是一项真实的、仔细测量的结果。它也是十六位开发者，在他们极其熟悉的仓库上，使用 2025 年初的工具——而且它不是那句平坦的句子“AI 让开发者变慢”。同一团队随后的数据已经指向了相反的方向。



每个人都预测 AI 会加快工作速度——开发者 -24%，机器学习专家 -38%，经济学家 -39%，以及开发者自己事后相信的 -20%。秒表发现 +19% 更慢，区间从 +2% 到 +39%。感觉与测量之间的差距才是发现。

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- not peer-reviewed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.

这项研究测量了什么——16 位专家级开源开发者，246 个真实任务，熟悉的仓库，2025 年初的工具，随机化——以及它没有测量什么：不是大多数开发者，不是其他领域，不是一条固定法则，未经同行评议。

Original diagram —The Clean Paper CC BY 4.0

这测量了什么，以及随机化试验为你买到了什么

随机对照试验（RCT）是医学用来区分真实效应和愿望效应的工具。在这里，246 个任务中的每一个都被随机分配为允许或不允许使用 AI，因此平均而言，两堆之间唯一的系统性差异就是 AI 本身。这就是让你能够说 AI 导致了时间变化，而不是仅仅注意到伸手使用 AI 的人恰好因为其他原因更快或更慢。这很重要，因为 AI 编程收益的通常证据——自我报告和基准分数——不能做到这一点：一个基准不是真实的工作，而一份自我报告，正如这项研究所显示的，可以自信地是错的。这里的开发者不是在新玩具上摸索的新手。他们是他们熟知的大型、成熟开源项目的确定贡献者，在使用这些工具方面有一些先前的经验。

作者做了什么

- 运行了一项**随机对照试验**（METR：Joel Becker、Nate Rush、Beth Barnes、David Rein）。十六位经验丰富的开源开发者，每个人都在他们定期贡献并熟知的大型仓库上工作——在具体项目上平均有五年经验。
- 使用了 **246 个真实任务**——从这些项目自己的问题追踪器中提取的错误修复、功能和重构。每个任务被随机分配到“允许 AI”或“禁止 AI”。
- “允许 AI”意味着 **2025 年初的工具：Cursor Pro 加 Claude 3.5/3.7 Sonnet**。主要指标是每个任务的实际完成时间。同时，团队收集了预测（从事前开发者那里）和估算（从事后他们那里），以及

来自经济学和机器学习的专家的预测。

他们发现了什么

- **使用 AI 时，任务用时多出 19%。**不是更快——更慢。95% 置信区间大致从 +2% 到 +39%，因此方向是确定的，即使确切大小不是。
- **每个人都预测了相反的结果。**开发者预测了 24% 的加速；机器学习专家约 38%；经济学家约 39%。三组人都预期 AI 能节省大量时间；秒表发现它花费了一些。
- **感知差距。**在完成工作并出来变慢之后，开发者仍然估计 AI 让他们加快了约 20%——他们感觉到的和时钟记录之间的差距约为 40 个百分点。
- **候选原因，权衡而非证明。**作者排列了可以解释减速的因素：这些开发者深度了解自己的代码库，因此助手能添加的较少；成熟项目带有很高的、通常是隐含的质量标准；仓库庞大且充满模型不具备的上下文；而真实时间花在了提示上，然后审查和修正 AI 输出。他们将这些作为线索呈现，而非判决。

这没有显示什么

- 它**没有**显示 AI 未能加速大多数开发者。作者直接这样说：十六位专家在他们了如指掌的代码上不是普通开发者在普通任务上的样子。
- 它**没有**显示 AI 是无用的，或者它在其他情境下让人们变慢——代码库的新来者、绿地工作、不熟悉的语言，或完全不同的领域。
- 它**没有**将工具冻结在时空中。这是 2025 年初的 Cursor 和 Claude 3.5/3.7 Sonnet；作者明确表示更好的工具，或使用这些工具的更好方式，即使在这种确切环境中也可能改变结果。
- 它是一篇**预印本**（2025 年 7 月发布，尚未经同行评议），作者指出他们不能完全排除实验假象——尽管该发现在他们的分析中经受了检验。
- 它**没有**授权那种令人安心的解读，即开发者一定以某种其他方式获得了——学到了更多、感觉更快乐、写出了更好的代码。这里测量的唯一一件事，感觉上的加速，正是数据所否定的。

证据有多强

- **设计异常诚实。**随机化，真实任务，真实仓库，实际计时——比大多数关于 AI 编程的声称所依赖的自我报告和基准排行榜上升了一大步。19% 的减速经受了作者的稳健性检查。
- **最具可移植性的发现是感知差距。**专家对自己 AI 加速的直觉错了大约 40 个百分点，且偏向乐观方向。这对每一份关于 AI 生产力提升的自我报告都是一个警示——包括这项研究自己的数字。
- 它是一个**快照，而非趋势**。METR 自己在 2026 年 2 月的后续研究，在同样类型的开发者上使用更新的工具，指向了加速——对于回访开发者约 -18%，对于新开发者约 -4%——但作者将其标记为弱证据，被谁愿意参与所扭曲（开发者越来越拒绝在没有 AI 的情况下工作，报酬率下降，任务选择偏斜）。诚实的解读是图景在移动，而即使是移动本身，在报告时也将拇指从秤上移开了。

为什么重要

简洁摘要

在一项随机对照试验中，METR 让十六位经验丰富的开源开发者在他们熟知的代码库上完成 246 个真实任务，随机一半允许使用 2025 年初的 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。开发者预期 AI 削减任务时间约 24%；结果完成时间反而增加了 19%——事后他们仍然相信 AI 让他们加快了约 20%。这个感知差距是研究最尖锐、最稳健的核心。但这是十六位开发者，一个狭窄的环境，以及一个固定的 2025 年初快照；作者明确表示它并未显示 AI 未能帮助大多数开发者，他们自己 2026 年的后续研究已偏向加速，带有其自身的保留。一项值得认真对待的仔细测量——不是对 AI 编程的判决。

关于 AI 和编程的争论大多在演示和氛围上运行：一段流畅的屏幕录像，一个自信的主张，一个反驳的白眼。罕见的是，也是这篇文章值得一读的，是有人跑了那个枯燥的实验——随机化，计时真实工作，然后问人们感觉如何。答案对两个阵营都不舒服。它戳破了 AI 统一地在困难、熟悉的代码上加速专家开发者的故事。它也戳破了整洁的反面——“AI 让开发者变慢，已被证明”——因为同一团队的新数据已经偏向另一边。最持久的教训也是最微小和最人性的：做这项工作的人感觉更快，而可测量地更慢。“感觉更快”不是它真正更快的证据。测量它。

来源

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/>