



The CLEAN PAPER

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

资深开发者觉得 AI 让自己更快，实际测下来却更慢——这是真发现，但不是对 AI 编程的最终判决

METR 的一项随机对照试验，让 16 名资深开源开发者在自己熟悉的代码库中完成 246 个真实工作项目；每个任务随机决定是否允许使用 2025 年初的 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。开发者原本预期 AI 能把完成时间缩短约 24%；结果反而增加了 19%。更有意思的是，做完后他们仍相信 AI 让自己快了约 20%。这种“感觉与实测之间的落差”是研究最锐利、也最稳健的核心。但样本只有 16 人、情境很窄，而且只是固定在 2025 年初工具状态的一张快照；作者明确表示，这不能证明 AI 对多数开发者没有帮助，而他们自己 2026 年的后续研究已经倾向显示加速，虽然也有自己的限制。

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

资深开发者觉得 AI 让自己更快，实际测下来却更慢——这是真发现，但不是对 AI 编程的最终判决

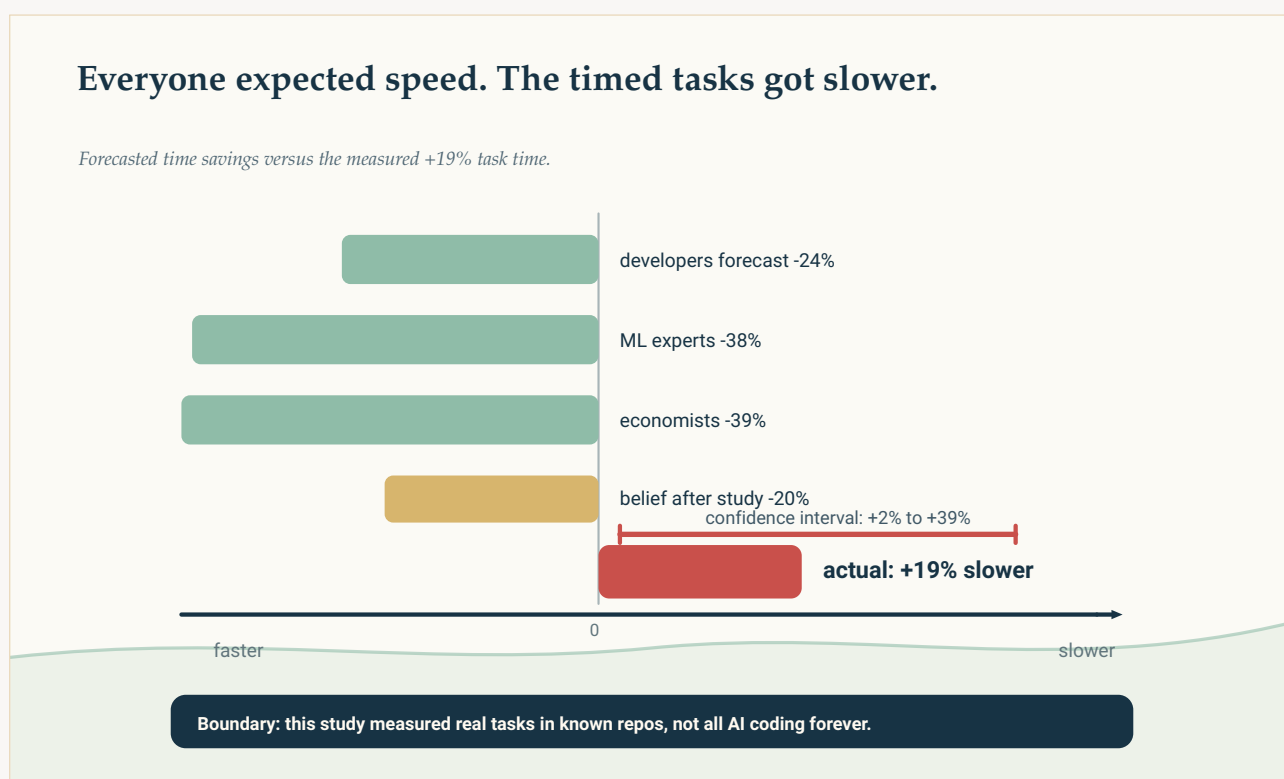
问一位有经验的程序员：“AI 编程助手有没有让你更快？”通常很快就会得到一个数字：省了 20%、30% 的时间。问经济学家或机器学习研究者，数字往往还会更大。

2025 年初，METR 的一个团队做了一件缓慢、昂贵，却真正能回答问题的事：直接测。他们找来 16 名资深开源开发者，交给他们 246 个来自大型、而且本人非常熟悉的代码库的真实任务；每一个任务都用随机方式决定能不能使用 AI 工具。然后计时。

开发者原本预测，AI 会把任务时间缩短约 24%。结果正好相反：允许 AI 的任务反而**多花了 19% 时间**。

而更值得停下来看的，是后面这件事：做完实验后，同一批开发者仍然相信 AI 让自己快了约 20%。他们实际更慢，感觉却更快；这两个数字之间的距离，才是这项研究最有意思的部分。

这是一个真实、仔细测量的结果。但它同时也只是 16 名开发者、在自己非常熟悉的代码库、使用 2025 年初工具的结果——而不是一句平面的“AI 让开发者变慢”。同一个团队较新的数据，甚至已经开始往相反方向走。



所有人都预测 AI 会加速工作：开发者 -24%、机器学习专家 -38%、经济学家 -39%，连开发者完成实验后的事后感受都还是 -20%。实际计时却显示 +19%——变慢——置信区间约 +2% 到 +39%。“感觉”与“测得”之间的落差，就是这项研究最值得带走的结果。

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- not peer-reviewed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.

这项研究真正测得的是什么：16 名资深开源开发者、246 个真实任务、熟悉的代码库、2025 年初工具、随机分配。它没有测得什么：不是多数开发者、不是其他领域、不是固定法则，也尚未同行评审。

Original diagram —The Clean Paper CC BY 4.0

这项研究量了什么，以及随机试验多给了你什么

随机对照试验（randomized controlled trial, RCT）是医学常用来区分“真正效果”与“看起来有希望”的工具。这里 246 个任务中的每一个，都被随机分配为可以或不可以用 AI；因此平均而言，两堆任务唯一系统性的差别就是 AI 本身。

这让你可以说，是 AI 造成了时间变化，而不是只观察到“比较喜欢用 AI 的人刚好本来就比较快或比较慢”。这点非常重要，因为 AI 软件开发增益常见的证据——自我报告与基准测试分数——做不到这件事：基准测试不是真实工作，而自我报告，正如这项研究展示的，可以很有自信地错。

这里的开发者也不是第一次碰工具的新手。他们是大型、成熟开源项目的既有贡献者，非常熟悉自己的代码库，而且已有一定 AI 工具使用经验。

作者做了什么

- METR 的 Joel Becker、Nate Rush、Beth Barnes、David Rein 执行一项**随机对照试验**。16 名资深开源开发者，每人都在自己长期参与、非常熟悉的大型代码库工作；平均已在对应项目工作约五年。
- 使用 **246 个真实任务**：来自那些项目 issue tracker 的 bug 修正、新功能与重构。每个任务随机分成“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**。主要指标是每个任

务真正花掉的完成时间。另外也收集开发者事前预测、事后估计，以及经济学与机器学习专家的预测。

他们发现了什么

- **使用 AI 时，任务反而多花 19% 时间。**不是更快，是更慢。95% 置信区间大约从 +2% 到 +39%，因此方向相当稳固，精确效应量则仍有不确定性。
- **所有人原本都猜错方向。**开发者预测快 24%，机器学习专家约 38%，经济学家约 39%。三群人都预期 AI 能省下大量时间；码表却显示它多花了一些。
- **感知落差。**做完工作、实际变慢后，开发者仍估计 AI 让自己快了约 20%——“感受到的”和“实际计时”之间，大约差 40 个百分点。
- **可能原因被列出、比较，但没有被证明。**作者提出几个可能解释：这些开发者对自己代码库太熟，助理能补充的东西较少；成熟项目有很高、而且常是默会的质量标准；代码库很大，模型缺少许多上下文；以及真实时间会消耗在提示、检查与修正 AI 输出。作者把这些当成线索，不是定论。

这并没有证明什么

- 它**没有**证明 AI 无法加速大多数开发者。作者直接这样说：16 名在自己熟悉代码上工作的专家，并不等于“平均开发者做平均任务”。
- 它**没有**证明 AI 没用，也没有证明它在其他情境会让人变慢——例如新加入某代码库的人、全新项目、不熟悉的语言，或其他领域。
- 它**没有**把工具永远冻结在原地。这是 2025 年初的 Cursor 与 Claude 3.5/3.7 Sonnet；作者明确表示，工具变好，或同样工具的使用方式变好，都可能让即使完全相同的情境得到不同结果。
- 这是一篇**预印本**（2025 年 7 月发布，尚未同行评审）；作者也表示无法完全排除实验设计造成的假象——虽然 19% 变慢的结果经过多种稳健性分析仍存在。
- 它**也不允许**一个安慰式解读：“也许开发者虽然慢一点，但学得更多、比较快乐，或代码更好。”这些可能性没有在这里被证明；研究真正测得的一件主观东西——“觉得自己变快”——恰好是数据直接反驳的。

证据有多强

- **研究设计异常诚实。**随机化、真实任务、真实代码库、实际计时——比 AI 软件开发讨论常依赖的自我报告与基准测试排行榜高出一大截。19% 变慢在作者的稳健性检查中仍然存在。
- **最可能跨情境成立的结果，是感知落差。**专家对“AI 让自己快多少”的直觉，往乐观方向错了约 40 个百分点。这对任何 AI 生产力自我报告都是警告——也包括这篇研究自己的受试者。
- **这是一张快照，不是一条趋势线。**METR 自己在 2026 年 2 月的后续研究，用类似开发者与较新工具，已经开始指向加速——很粗略地说，回锅开发者约 -18%，新加入者约 -4%。但作者自己把它标成弱证据，因为参与者选择偏误变得很严重：越来越多开发者拒绝在不能用 AI 的条件下工作，

报酬水平下降，而且任务选择也偏了。最诚实的读法是：图景正在移动，而且连这个移动本身都必须小心测。

为什么重要

AI 与软件开发的大部分争论，都靠 demo 和感觉运转：一段流畅的屏幕录影、一个很有信心的宣称，再配上一个反方向的白眼。真正少见、也让这篇值得看的，是有人真的去做了那个无聊的实验：随机分配、计时真实工作，再问大家感觉如何。

答案让两边都不太舒服。它戳破了“AI 一定能让熟悉复杂代码的专家全面暴增速度”的故事；也戳破了另一个整齐的相反故事——“已证明 AI 让开发者变慢”——因为同一团队的新数据已经开始往相反方向走。

最持久的教训反而最小、也最人性：做工作的人感觉更快，但测出来更慢。“感觉更快”不是它真的更快的证据。量它。

简明总结

在一项随机对照试验中，METR 让 16 名资深开源开发者完成 246 个来自熟悉代码库的真实任务，其中一半随机允许使用 2025 年初 AI 工具（Cursor Pro 加 Claude 3.5/3.7 Sonnet）。开发者原本预期 AI 能把任务时间缩短约 24%；结果反而让完成时间增加 19%，而做完后他们仍相信 AI 让自己快了约 20%。这个感知落差，是研究最锐利也最稳健的核心。但样本只有 16 人、情境很窄，而且只是 2025 年初工具状态的一张快照；作者明确表示，这不能证明 AI 对多数开发者没有帮助，而他们 2026 年的后续研究甚至已经倾向显示加速，只是也有自己的限制。这是一个值得认真看待的仔细测量，不是对 AI 软件开发的最终判决。

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/>