The Clean Paper

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

Des développeurs expérimentés se sentaient plus rapides avec l'IA tout en travaillant mesurablement plus lentement — le vrai résultat, et ce qu'il ne dit pas sur le code avec IA

Un essai randomisé contrôlé de METR a demandé à seize développeurs open source expérimentés d'effectuer 246 vraies tâches sur des bases de code qu'ils connaissaient bien, avec des outils d'IA du début 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) autorisés sur une moitié choisie au hasard. Ils s'attendaient à ce que l'IA réduise le temps de tâche d'environ 24 % ; elle l'a au contraire augmenté de 19 % — et après coup ils croyaient encore qu'elle les avait accélérés d'environ 20 %. Cet écart de perception est le cœur net et robuste de l'étude. Mais il s'agit de seize développeurs, d'un cadre étroit, et d'un instantané fixe du début 2025 : les auteurs sont explicites, cela ne montre pas que l'IA échoue à aider la plupart des développeurs, et leur propre suivi de 2026 pointe déjà vers un gain de vitesse, avec ses propres réserves.

Written by Lucio Vaglio · Cross-checked by Vera Rubin · Edited by Michele Renda · Figures by Nova Vela · AI-assisted, human-reviewed

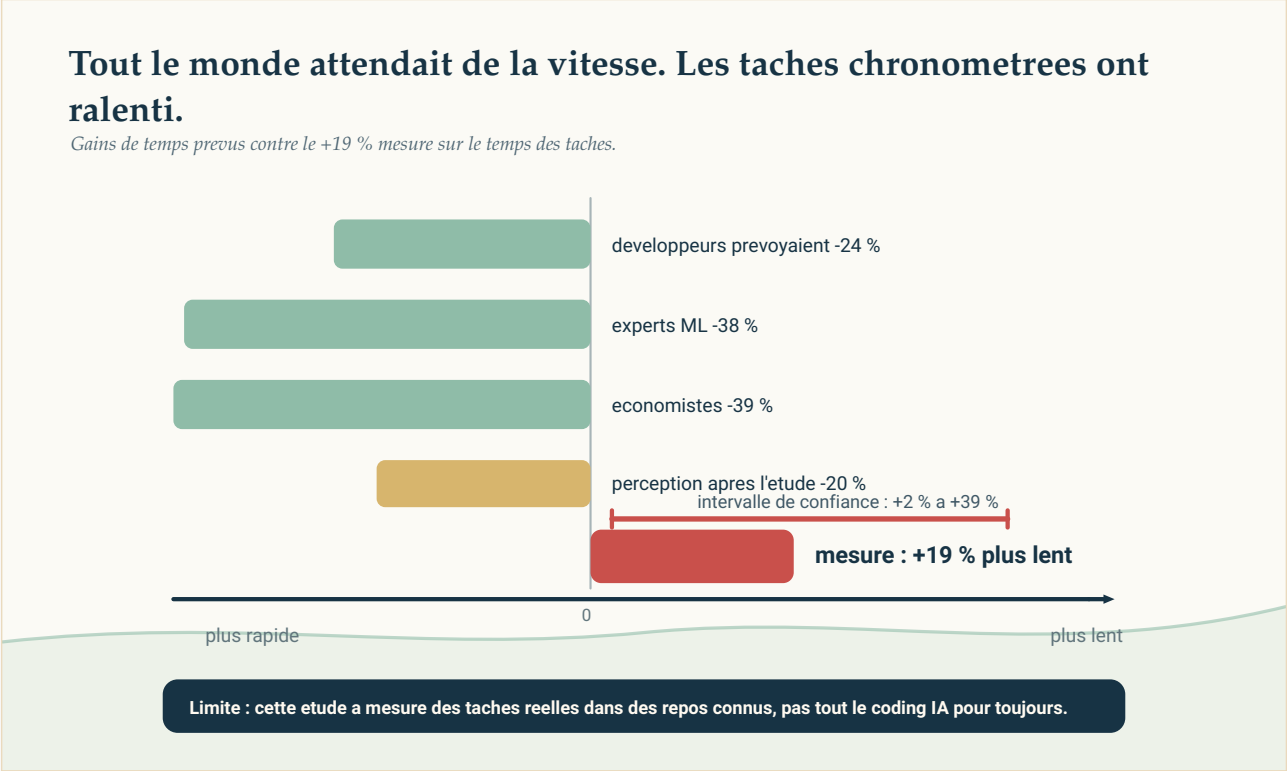
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

Des développeurs expérimentés se sentaient plus rapides avec l'IA tout en travaillant mesurablement plus lentement — le vrai résultat, et ce qu'il ne dit pas sur le code avec IA

Demandez à un programmeur expérimenté si un assistant de code par IA le rend plus rapide et vous obtiendrez souvent un chiffre : vingt, trente pour cent de temps gagné. Demandez à un économiste, ou à un chercheur en apprentissage automatique, et le chiffre grossit. Début 2025, une équipe de METR a fait la chose lente, coûteuse, et a vérifié. Elle a recruté seize développeurs open source aguerris, leur a confié 246 vraies tâches issues de grands dépôts qu'ils connaissaient bien, et — par tirage au sort, tâche par tâche — les a soit autorisés à utiliser des outils d'IA, soit non. Puis elle a chronométré le travail.

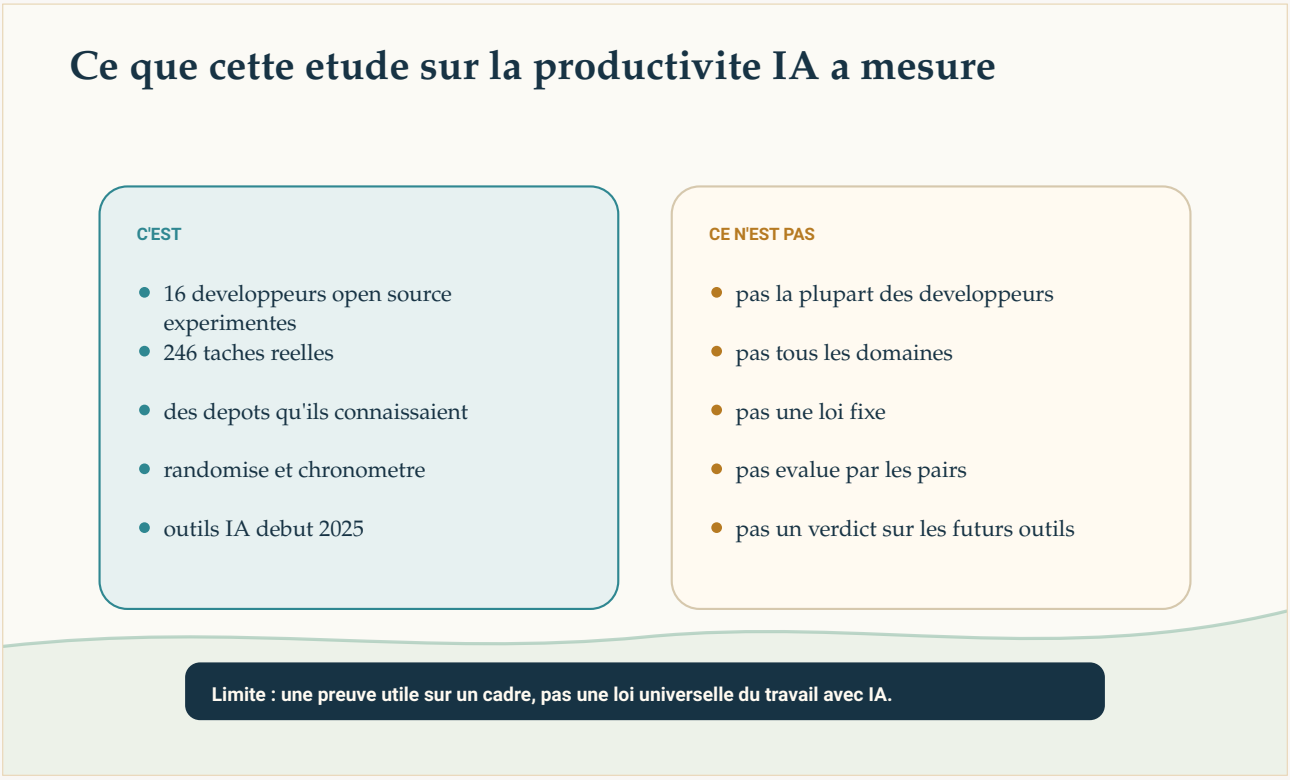
Les développeurs avaient prévu que l'IA réduirait leur temps de tâche d'environ 24 %. Elle a fait l'inverse : les tâches réalisées avec IA ont pris **19 % de plus**. Et voici la partie qui mérite qu'on s'y arrête : après avoir terminé, les mêmes développeurs croyaient encore que l'IA les avait accélérés, d'environ 20 %. Ils étaient plus lents, ils se sentaient plus rapides, et l'écart entre ces deux chiffres est la chose la plus intéressante de l'étude.

C'est un résultat réel, mesuré avec soin. C'est aussi seize développeurs, sur des dépôts qu'ils connaissent intimement, avec les outils du début 2025 — et ce n'est pas la phrase plate "l'IA ralentit les développeurs". Les données plus récentes de la même équipe pointent déjà dans l'autre sens.



Tout le monde prévoyait que l'IA accélérerait le travail — développeurs −24 %, experts ML −38 %, économistes −39 %, et la perception après coup des développeurs −20 %. Le chronomètre a trouvé +19 %

plus lent, avec un intervalle de +2 % à +39 %. L'écart entre ressenti et mesuré est le résultat.



Ce que l'étude mesure — 16 développeurs open source experts, 246 tâches réelles, des dépôts familiers, des outils du début 2025, une randomisation — et ce qu'elle ne mesure pas : pas la plupart des développeurs, pas les autres domaines, pas une loi fixe, pas une étude évaluée par les pairs.

Ce qui a été mesuré, et ce qu'un essai randomisé apporte

Un **essai randomisé contrôlé (RCT)** est l'outil que la médecine utilise pour distinguer un effet réel d'un effet espéré. Ici, chacune des 246 tâches a été assignée au hasard à une condition avec IA autorisée ou sans IA, de sorte qu'en moyenne la seule différence systématique entre les deux piles soit l'IA elle-même. C'est ce qui permet de dire que l'IA a *causé* le changement de temps, au lieu de seulement constater que les personnes qui se tournent vers l'IA sont plus rapides ou plus lentes pour d'autres raisons. C'est important parce que les preuves habituelles de gains de productivité en codage avec IA — auto-déclarations et scores de benchmark — ne peuvent pas faire cela : un benchmark n'est pas du vrai travail, et une auto-déclaration, comme cette étude le montre, peut être fausse avec assurance. Les développeurs ici n'étaient pas des novices découvrant un nouveau jouet. C'étaient des contributeurs établis de grands projets open source matures qu'ils connaissaient bien, avec une expérience préalable des outils.

Ce que les auteurs ont fait

- Ils ont mené un **essai randomisé contrôlé** (METR : Joel Becker, Nate Rush, Beth Barnes, David Rein). Seize développeurs open source expérimentés, chacun travaillant sur un grand dépôt auquel il contribue régulièrement et qu'il connaît bien — en moyenne cinq ans sur les projets concernés.
- Ils ont utilisé **246 vraies tâches** — corrections de bugs, fonctionnalités et refactorings tirés des propres trackers d'issues de ces projets. Chaque tâche a été assignée au hasard à "IA autorisée" ou "IA interdite".
- "IA autorisée" signifiait **outillage du début 2025 : Cursor Pro avec Claude 3.5/3.7 Sonnet.** La mesure principale était le temps réel nécessaire pour terminer chaque tâche. En parallèle, l'équipe a recueilli des prévisions (des développeurs avant le travail) et des estimations (après), ainsi que des prédictions d'experts en économie et en apprentissage automatique.

Ce qu'ils ont trouvé

- **Avec l'IA, les tâches ont pris 19 % de plus.** Pas moins — plus. L'intervalle de confiance à 95 % va d'environ +2 % à +39 %, donc la direction est solide même si la taille exacte ne l'est pas autant.
- **Tout le monde avait prédit l'inverse.** Les développeurs prévoyaient un gain de 24 % ; les experts en apprentissage automatique environ 38 % ; les économistes environ 39 %. Les trois groupes s'attendaient à ce que l'IA fasse gagner beaucoup de temps ; le chronomètre a trouvé qu'elle en coûtait.
- **L'écart de perception.** Après avoir fait le travail et être sortis plus lents, les développeurs estimaient encore que l'IA les avait accélérés d'environ 20 % — un écart d'environ 40 points entre ce qu'ils ressentaient et ce que l'horloge enregistrait.
- **Des causes candidates, pesées plutôt que prouvées.** Les auteurs alignent des facteurs qui pourraient expliquer le ralentissement : ces développeurs connaissent très bien leurs propres bases de code, donc il y a moins à ajouter pour un assistant ; les projets matures portent des standards de qualité élevés et souvent implicites ; les dépôts sont grands et pleins de contexte qu'un modèle n'a pas ; et du vrai temps part dans le prompt, puis dans la revue et la correction de la sortie de l'IA. Ils présentent cela comme des pistes, pas comme des verdicts.

Ce que cela ne montre pas

- Cela ne montre **pas** que l'IA échoue à accélérer la plupart des développeurs. Les auteurs le disent directement : seize experts sur du code qu'ils connaissent par cœur ne sont pas le développeur moyen sur la tâche moyenne.
- Cela ne montre **pas** que l'IA est inutile, ni qu'elle ralentit les gens dans d'autres contextes — nouveaux venus sur une base de code, travail greenfield, langages inconnus, ou autres domaines.
- Cela ne fige **pas** les outils. Il s'agit de Cursor et Claude 3.5/3.7 Sonnet début 2025 ; les auteurs sont

explicites : de meilleurs outils, ou de meilleures façons d'utiliser ceux-ci, pourraient changer le résultat même dans ce cadre exact.
- C'est un **prépublication** (mis en ligne en juillet 2025, pas encore évalué par les pairs), et les auteurs notent qu'ils ne peuvent pas exclure entièrement des artefacts expérimentaux — même si le résultat tient dans leurs analyses.
- Cela n'autorise **pas** la lecture rassurante selon laquelle les développeurs auraient forcément gagné autre chose — appris davantage, été plus heureux, écrit du meilleur code. La seule chose mesuree ici, le gain de vitesse ressenti, est précisément ce que les données contredisent.

Quelle est la force de la preuve

- **Le design est inhabituellement honnête.** Randomisation, vraies tâches, vrais dépôts, chronométrage réel — un gros progrès par rapport aux auto-déclarations et aux classements de benchmarks sur lesquels reposent la plupart des affirmations à propos du code avec IA. Le ralentissement de 19 % a résisté aux contrôles de robustesse des auteurs.
- **Le résultat le plus transférable est l'écart de perception.** L'intuition experte sur son propre gain de vitesse avec l'IA était fausse d'environ 40 points, dans le sens optimiste. C'est un avertissement pour *tout* gain de productivité IA auto-déclaré — y compris les chiffres de cette étude.
- **C'est un instantané, pas une tendance.** Le suivi de février 2026 de METR, avec le même type de développeurs et des outils plus récents, pointe vers un gain de vitesse — très approximativement -18 % pour les développeurs revenant dans l'étude et -4 % pour les nouveaux — mais les auteurs le décrivent comme une preuve faible, déformée par les personnes qui ont accepté de participer (les développeurs refusaient de plus en plus de travailler sans IA, le tarif de rémunération a baissé, et la sélection des tâches a dérivé). La lecture honnête est que l'image bouge, et que même ce mouvement est rapporté sans appuyer sur la balance.

Pourquoi c'est important

La plupart des débats sur l'IA et la programmation reposent sur des démos et des impressions : une capture vidéo fluide, une affirmation confiante, un soupir opposé. Ce qui est rare, et ce qui rend cette étude intéressante, c'est que quelqu'un a mené l'expérience ennuyeuse — randomiser, chronométrer du vrai travail, puis demander aux gens comment cela s'était passé. La réponse est inconfortable pour les deux camps. Elle perce le récit selon lequel l'IA turbocharge uniformément des développeurs experts sur du code difficile et familier. Et elle perce aussi l'opposé trop net — "l'IA ralentit les développeurs, c'est prouvé" — parce que les données plus récentes de la même équipe penchent déjà dans l'autre sens. La leçon la plus durable est aussi la plus petite et la plus humaine : les personnes qui faisaient le travail se sentaient plus rapides tout en étant mesurablement plus lentes. "Ça semble plus rapide" n'est pas une preuve que ça l'est. Mesurez.

Résumé net

Dans un essai randomisé contrôlé, METR a demandé à seize développeurs open source expérimentés de terminer 246 vraies tâches sur des bases de code qu'ils connaissaient bien, avec des outils d'IA du début 2025 (Cursor Pro plus Claude 3.5/3.7 Sonnet) autorisés sur une moitié choisie au hasard. Les développeurs s'attendaient à ce que l'IA réduise leur temps de tâche d'environ 24 % ; elle l'a au contraire augmenté de 19 % — et après coup ils croyaient encore qu'elle les avait accélérés d'environ 20 %. Cet écart de perception est le cœur net et robuste de l'étude. Mais il s'agit de seize développeurs, d'un cadre étroit, et d'un instantané fixe du début 2025 ; les auteurs sont explicites : cela ne montre pas que l'IA échoue à aider la plupart des développeurs, et leur propre suivi de 2026 pointe déjà vers un gain de vitesse, avec ses propres réserves. Une mesure prudente à prendre au sérieux — pas un verdict sur le code avec IA.

Sources

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/