



The CLEAN PAPER

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

นักพัฒนามากประสบการณ์รู้สึกว่ AI ทำให้เร็วขึ้น ทั้งที่วัด จริงแล้วช้าลง — นี่คือผลที่แท้จริง และไม่ใช่คำตัดสินเรื่อง AI coding

การทดลองแบบสุ่มมีกลุ่มควบคุมของ METR ให้นักพัฒนาโอเพนซอร์สมากประสบการณ์ 16 คนทำงานจริง 246 งานบน codebase ที่รู้จักดี โดยสุ่มอนุญาตเครื่องมือ AI ต้นปี 2025 (Cursor Pro ร่วมกับ Claude 3.5/3.7 Sonnet) ในครึ่งหนึ่งของงาน พวกเขาคาดว่า AI จะลดเวลาได้ราว 24%; แต่จริง ๆ เวลาทำงานเพิ่ม 19% — และสงสัยบยังเชื่อว่า AI ทำให้เร็วขึ้นราว 20% ช่องว่างด้านการรับรู้คือแก่นที่ชัดและแข็งแกร่งที่สุดของงาน แต่ตัวอย่างมีเพียง 16 คน บริบทแคบ และเป็นภาพถ่ายของต้นปี 2025: ผู้เขียนระบุชัดว่ามันไม่ได้แสดงว่า AI ช่วยนักพัฒนาส่วนใหญ่ไม่ได้ และงานติดตามปี 2026 ของทีมเดียวกันเริ่มเอนไปทางการเพิ่มความเร็ว พร้อมข้อจำกัดของมันเอง

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

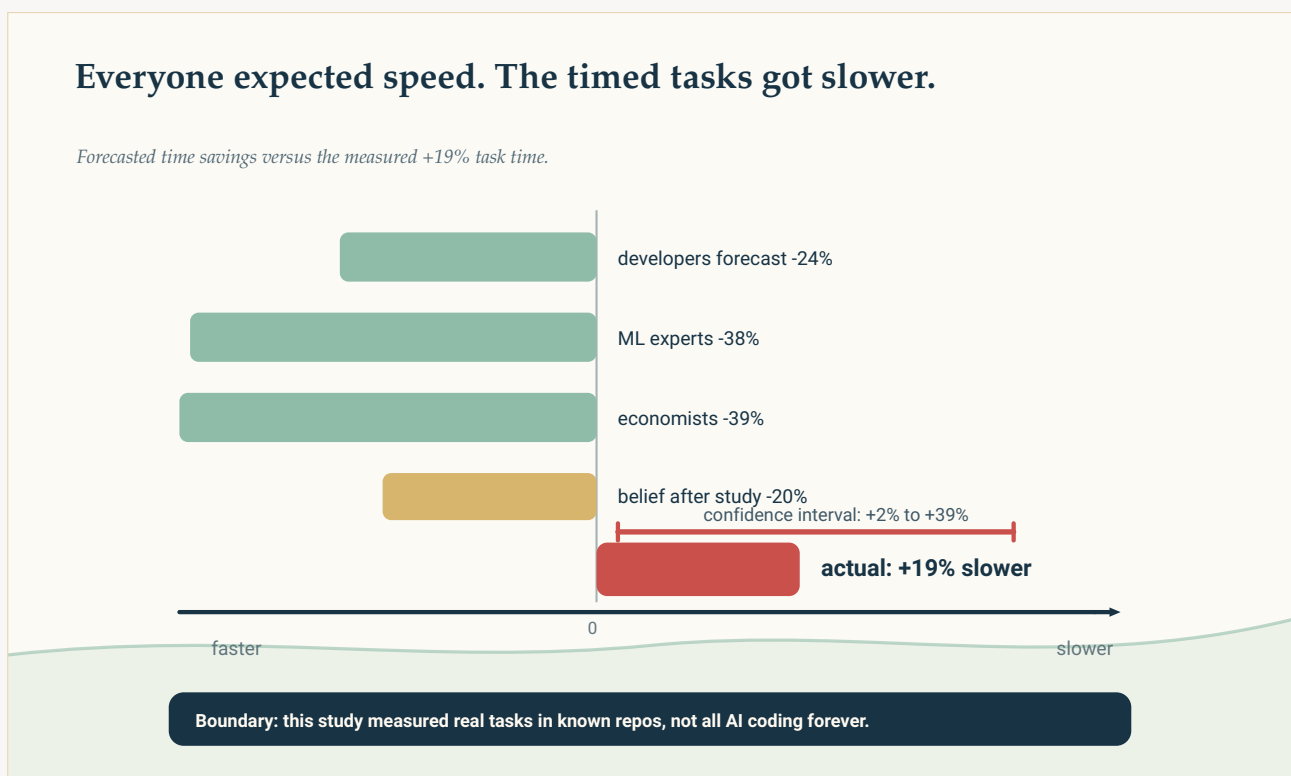
Paper: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Joel Becker, Nate Rush, Beth Barnes, David Rein (METR), arXiv:2507.09089 [cs.AI] (preprint) · DOI: 10.48550/arXiv.2507.09089

นักพัฒนามากประสบการณ์รู้สึกว่ AI ทำให้เร็วขึ้น ทั้งที่วัดจริงแล้วช้าลง — นี่คือน้ื่องานวิจัยพบ และสิ่งที่มันไม่ได้ตัดสินเกี่ยวกับการเขียนโค้ดด้วย AI

ลองถามโปรแกรมเมอร์ที่มีประสบการณ์ว่าผู้ช่วยเขียนโค้ดด้วย AI ทำให้เขาทำงานเร็วขึ้นไหม คุณมักจะได้อำตอบเป็นตัวเลข: ประหยัดเวลาให้ผมสักยี่สิบ สามสิบเปอร์เซ็นต์ ถ้าไปถามนักเศรษฐศาสตร์หรือนักวิจัยแมชชีนเลิร์นนิง ตัวเลขมักยิ่งสูงขึ้น ต้นปี 2025 ทีมหนึ่งที่ METR เลือกทำสิ่งที่ช้าและแพงกว่า คือวัดจริง พวกเขาให้นักพัฒนาโอเพนซอร์สระดับชำนาญ 16 คนทำงานจริง 246 งานจาก codebase ขนาดใหญ่ที่แต่ละคนรู้จักดี และในแต่ละงานสุ่มรวมกับโยนเหรียญว่าจะอนุญาตให้ใช้เครื่องมือ AI หรือไม่ จากนั้นจับเวลาการทำงาน

นักพัฒนาคาดไว้ว่า AI จะลดเวลาทำงานลงประมาณ 24% แต่ผลกลับตรงกันข้าม: งานที่ทำโดยมี AI ใช้เวลา **นานขึ้น 19%** และส่วนที่น่าคิดที่สุดคือ หลังทำเสร็จ นักพัฒนากลุ่มเดิมยังเชื่อว่า AI ทำให้ตัวเองเร็วขึ้นประมาณ 20% พวกเขาทำงานช้าลง แต่รู้สึกว่เร็วขึ้น และระยะห่างระหว่างสองตัวเลขนี่คือน้ื่องานที่น่าสนใจที่สุดในงานวิจัย

นี่เป็นผลที่วัดจริงและวัดอย่างระมัดระวัง แต่ก็เป็ผลจากนักพัฒนาเพียง 16 คน ที่ทำงานบน repository ที่ตัวเองคุ้นเคยอย่างลึกซึ้ง ด้วยเครื่องมือของต้นปี 2025 — มันจึงไม่ใช่ประโยชน์สั้น ๆ ว่ “AI ทำให้นักพัฒนาช้าลง” ข้อมูลรุ่นใหม่กว่าของทีมเดียวกันเองเริ่มเข้าไปอีกทางแล้ว



ทุกกลุ่มคาดว่า AI จะทำให้งานเร็วขึ้น — นักพัฒนา -24%, ผู้เชี่ยวชาญ ML -38%, นักเศรษฐศาสตร์ -39% และแม้แต่ความเชื่อของนักพัฒนาหลังจบการทดลองก็อยู่ที่ -20% แต่การจับเวลาพบว่าช้าลง +19% โดยช่วงประมาณการอยู่ที่ +2% ถึง +39% ช่องว่างระหว่างความรู้สึกกับสิ่งที่วัดได้คือผลลัพธ์สำคัญ

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.

สิ่งที่งานวิจัยวัด — นักพัฒนาโอเพนซอร์สผู้เชี่ยวชาญ 16 คน งานจริง 246 งาน repository ที่คุ้นเคย เครื่องมือ AI ต้นปี 2025 และการสุ่มเงื่อนไข — กับสิ่งที่มันไม่ได้วัด: ไม่ใช่นักพัฒนาส่วนใหญ่ ไม่ใช่ทุกสาขา ไม่ใช่กฎหมายตายตัว และยังไม่ผ่านการประเมินโดยผู้ทรงคุณวุฒิ

Original diagram — The Clean Paper CC BY 4.0

งานนี้วัดอะไร และการทดลองแบบสุ่มช่วยอะไร

การทดลองแบบสุ่มมีกลุ่มควบคุม (แบบสุ่มมีกลุ่มควบคุม (randomized controlled trial; RCT)) เป็นเครื่องมือที่วงการแพทย์ใช้แยกผลจริงออกจากผลที่เราเพียงหวังว่าจะมี ในงานนี้ งานทั้ง 246 งานถูกสุ่มให้ทำในเงื่อนไขที่อนุญาต AI หรือไม่อนุญาต AI ดังนั้นโดยเฉลี่ยแล้ว ความแตกต่างอย่างเป็นระบบระหว่างสองกลุ่มควรเหลือเพียงเรื่อง AI นั่นทำให้เราพูดได้ว่า AI *เป็นสาเหตุ* ของการเปลี่ยนแปลงด้านเวลา แทนที่จะเพียงสังเกตว่าคนที่เลือกหยิบ AI มาใช้บังเอิญเร็วหรือช้ากว่าด้วยเหตุอื่น เรื่องนี้สำคัญ เพราะหลักฐานที่มักใช้สนับสนุนว่า AI ช่วยเขียนโค้ดได้เร็วขึ้น — การรายงานด้วยตนเองและคะแนนจากชุดทดสอบมาตรฐาน — ไม่สามารถทำแบบนี้ได้: ชุดทดสอบมาตรฐาน ไม่ใช่งานจริง และการประเมินตัวเอง อย่างที่งานนี้แสดง อาจผิดได้อย่างมั่นใจ นักพัฒนาในงานนี้ไม่ใช้มือใหม่ที่กำลังมกกับของเล่นใหม่ พวกเขาเป็น contributor ประจำของโครงการโอเพนซอร์สขนาดใหญ่ที่พัฒนาเต็มที่แล้ว รู้จักโค้ดดี และเคยใช้เครื่องมือเหล่านั้นมาบ้าง

ผู้วิจัยทำอะไร

- ทำ **การทดลองแบบสุ่มมีกลุ่มควบคุม** (METR: Joel Becker, Nate Rush, Beth Barnes, David Rein) กับนักพัฒนาโอเพนซอร์สภาคประชาการณ์ 16 คน แต่ละคนทำงานกับ repository ขนาดใหญ่ที่ตัวเองมีส่วนร่วมเป็นประจำและรู้จักดี — โดยเฉลี่ยอยู่กับโครงการนั้นมาราวห้าปี
- ใช้ **งานจริง 246 งาน** — การแก้บั๊ก เพิ่มฟีเจอร์ และ refactor ที่นำมาจาก issue tracker ของโครงการเหล่านั้นเอง แต่ละงานถูกสุ่มเป็น “อนุญาต AI” หรือ “ไม่อนุญาต AI”

- “อนุญาต AI” หมายถึง **เครื่องมือต้นปี 2025: Cursor Pro ร่วมกับ Claude 3.5/3.7 Sonnet** ตัวชี้วัดหลักคือเวลาที่ใช้จริงจนงานเสร็จ นอกจากนี้ทีมยังเก็บคำคาดการณ์ของนักพัฒนาก่อนทำงาน การประเมินหลังทำงาน และคำทำนายจากผู้เชี่ยวชาญด้านเศรษฐศาสตร์และแมชชีนเลิร์นนิง

พวกเขาพบอะไร

- **เมื่อใช้ AI งานใช้เวลาเพิ่มขึ้น 19%** ไม่ได้เร็วขึ้น แต่ช้าลง ช่วงความเชื่อมั่น 95% อยู่ราว +2% ถึง +39% ดังนั้นทิศทางของผลค่อนข้างชัด แม้นาฬิกาที่แน่นอนยังไม่ชัดเจนเท่ากัน
- **ทุกกลุ่มทำนายตรงข้าม** นักพัฒนาคาดว่าเร็วขึ้น 24%; ผู้เชี่ยวชาญแมชชีนเลิร์นนิงราว 38%; นักเศรษฐศาสตร์ราว 39% ทั้งสามกลุ่มคาดว่า AI จะประหยัดเวลาได้มาก แต่การจับเวลาพบว่ามันกินเวลาเพิ่ม
- **ช่องว่างด้านการรับรู้** แม้ทำงานออกมาช้าลง นักพัฒนายังประเมินว่า AI ทำให้เร็วขึ้นประมาณ 20% — ต่างกันราว 40 จุดเปอร์เซ็นต์ระหว่างสิ่งที่รู้สึกกับสิ่งที่นาฬิกาวัดได้
- **เหตุผลที่เป็นไปได้ ซึ่งผู้เขียนยังไม่เห็นแต่ไม่ได้พิสูจน์** นักพัฒนากลุ่มนี้รู้จัก codebase ของตัวเองอย่างลึก จึงมีพื้นที่ให้ผู้ช่วยเพิ่มคุณค่าน้อยกว่า; โครงการที่โตเต็มที่มีมาตรฐานคุณภาพสูงและมักไม่ได้เขียนไว้ทั้งหมด; repository ใหญ่และมีบริบทมากที่โมเดลไม่รู้; และเวลาใช้งานจริงต้องเสียไปกับการเขียน พรอมป์ รวมถึงตรวจและแก้ผลลัพธ์จาก AI ผู้เขียนเสนอสิ่งเหล่านี้เป็นเบาะแส ไม่ใช่คำตอบตัดสิน

สิ่งที่ผลนี้ไม่ได้แสดง

- มัน **ไม่ได้แสดงว่า AI ทำให้ให้นักพัฒนาส่วนใหญ่เร็วขึ้นไม่ได้** ผู้เขียนพูดตรง ๆ ว่า ผู้เชี่ยวชาญ 16 คนที่ทำงานกับโค้ดซึ่งตัวเองรู้แทบขึ้นใจ ไม่ใช่ให้นักพัฒนาโดยเฉลี่ยที่ทำงานโดยเฉลี่ย
- มัน **ไม่ได้แสดงว่า AI ไร้ประโยชน์** หรือว่าทำให้คนช้าลงในบริบทอื่น — เช่น คนที่เพิ่งเข้ามาใน codebase งาน green-field ภาษาโปรแกรมที่ไม่คุ้นเคย หรือสาขาอื่นทั้งหมด
- มัน **ไม่ได้ตรงเครื่องมือไว้ตลอดกาล** นี่คือ Cursor กับ Claude 3.5/3.7 Sonnet ของต้นปี 2025 ผู้เขียนระบุชัดว่าเครื่องมือที่ดีขึ้น หรือวิธีใช้เครื่องมือเดิมที่ดีขึ้น อาจเปลี่ยนผลลัพธ์แม้ในบริบทเดียวกันนี้ได้
- งานนี้เป็น **บทความก่อนตีพิมพ์** (โพสต์เดือนกรกฎาคม 2025 และยังไม่ผ่าน การประเมินโดยผู้ทรงคุณวุฒิ) และผู้เขียนยอมรับว่ายังตัดความเป็นไปได้ของ artifact จากการทดลองออกไม่ได้ทั้งหมด แม้ผลหลักจะคงอยู่ในการวิเคราะห์หลายแบบ
- มัน **ไม่ได้เปิดทางให้ตีความปลอมใจ** ว่านักพัฒนาอาจได้ประโยชน์ทางอื่นแน่ ๆ — เรียนรู้มากขึ้น มีความสุขขึ้น หรือเขียนโค้ดดีขึ้น สิ่งหนึ่งที่วัดตรง ๆ คือความรู้สึกว่าเร็วขึ้น และข้อมูลกลับขัดแย้งกับความรู้สึกนั้นโดยตรง

หลักฐานเชิงแรงแคไหน

- **การออกแบบข้อตรงต่อคำถามอย่างผิดปกติ** มีการสุ่ม ใช้งานจริง คลังโค้ดจริง และจับเวลาจริง — ก้าวใหญ่จากงานที่อาศัยการรายงานด้วยตนเองหรืออันดับจากชุดทดสอบมาตรฐาน ซึ่งเป็นฐานของข้ออ้างจำนวนมากเรื่อง AI coding ผลช้าลง 19% ยังอยู่หลังการทดสอบความทนทานของผลหลายแบบ
- **ผลที่น่าจะใช้ได้กว้างที่สุดคือช่องว่างด้านการรับรู้** สัญชาตญาณของผู้เชี่ยวชาญเกี่ยวกับว่าตัวเองเร็วขึ้นเพราะ AI เท่าไร ผิดไปประมาณ 40 จุดในทิศทางที่มองโลกในแง่ดี นี่เป็นคำเตือนต่อ *ทุก* ตัวเลขเพิ่มผลิตภาพจาก AI ที่มาจากการประเมินตนเอง — รวมถึงตัวเลขของงานนี้เอง

- **นี่เป็นภาพถ่าย ณ ช่วงเวลา ไม่ใช่แนวโน้มถาวร** งานติดตามของ METR ในเดือนกุมภาพันธ์ 2026 ซึ่งให้นักพัฒนาประเภทใกล้เคียงกันแต่เครื่องมือใหม่กว่า เริ่มชี้ไปที่การทำงานเร็วขึ้น — คร่าว ๆ -18% สำหรับนักพัฒนาที่กลับมาทำอีกครั้ง และ -4% สำหรับผู้เข้าร่วมใหม่ — แต่ผู้เขียนเองระบุว่านี่เป็นหลักฐานอ่อน เพราะถูกบิดเบือนจากคนที่ยอมเข้าร่วม (นักพัฒนาปฏิเสธทำงานโดยไม่มี AI มากขึ้น อัตราค่าตอบแทนลดลง และการเลิกงานเอนเอียง) การอ่านที่ซื่อสัตย์ที่สุดคือ ภาพกำลังเปลี่ยน และแม้แต่หลักฐานของการเปลี่ยนนั้นก็นำเสนอพร้อมข้อจำกัดอย่างชัดเจน

ทำไมจึงสำคัญ

ข้อถกเถียงเรื่อง AI กับการเขียนโปรแกรมส่วนใหญ่ขับเคลื่อนด้วยเดโมและความรู้สึก: วิดีโอหน้าจอที่ดูไหลลื่น คำกล่าวอย่างมั่นใจ แล้วก็เสียงกลองตอกจากอีกฝั่ง สิ่งที่เราได้ยาก และทำให้งานนี้ควรค่าแก่การอ่าน คือมีคนยอมทำการทดลองที่น่าเบื่อจริง ๆ — สุ่มเงื่อนไข จับเวลางานจริง แล้วถามคนทำงานว่ารู้สึกอย่างไร คำตอบทำให้ทั้งสองค่ายไม่สบายใจ มันเจาะเรื่องแล้วว่า AI ทำให้ผู้เชี่ยวชาญทุกคนพุ่งเร็วขึ้นอย่างสม่ำเสมอจนได้ยากที่คุณเคย และในเวลาเดียวกันก็เจาะข้อสรุปตรงข้ามแบบเรียบร้อยเกินจริงว่า “AI ทำให้นักพัฒนาช้าลง พิสูจน์แล้ว” เพราะข้อมูลใหม่กว่าของทีมนี้อย่างเดียวก็เริ่มเอนไปอีกทาง บทเรียนที่ทนที่สุดกลับเป็นเรื่องเล็กและเป็นมนุษย์ที่สุด: คนทำงานรู้สึกว่าร็วขึ้น ทั้งที่วัดแล้วช้าลง “รู้สึกว่าร็วขึ้น” ไม่ใช่หลักฐานว่าร็วขึ้นจริง วัดมัน

สรุป

ในการทดลองแบบสุ่มมีกลุ่มควบคุม METR ให้นักพัฒนาโอเพนซอร์สมากประสบการณ์ 16 คนทำงานจริง 246 งานบน code-base ที่พวกเขารู้จักดี โดยสุ่มให้ครึ่งหนึ่งของงานใช้เครื่องมือ AI ต้นปี 2025 (Cursor Pro ร่วมกับ Claude 3.5/3.7 Sonnet) ได้ นักพัฒนาคาดว่า AI จะลดเวลาทำงานประมาณ 24%; แต่จริง ๆ แล้วเวลาทำงานเพิ่ม 19% — และหลังจากนั้นพวกเขายังเชื่อว่า AI ทำให้ตัวเองเร็วขึ้นประมาณ 20% ช่องว่างด้านการรับรู้ก็คือแก่นที่ชัดและแข็งแกร่งที่สุดของงาน แต่ตัวอย่างมีเพียง 16 คนบริบทแคบ และเป็นภาพถ่ายของต้นปี 2025 ผู้เขียนระบุชัดว่ามันไม่ได้แสดงว่า AI ช่วยนักพัฒนาส่วนใหญ่ไม่ได้ และงานติดตามปี 2026 ของพวกเขาเองเริ่มชี้ไปที่การเพิ่มความเร็ว พร้อมข้อจำกัดของมันเอง นี่คือการวัดที่ควรเอาจริง — ไม่ใช่คำตัดสินสุดท้ายเรื่องการเขียนโค้ดด้วย AI

แหล่งข้อมูล

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/>