The Fast and Spurious

Developer Productivity with GenAI

Sadia Afroz
Oregon State University

Zixuan (Steve) Feng [1]
OSU → VCU

Tyler Menezes [2]
CodeDay

Katie Kimura
Oregon State University

Bianca Trinkenreich
Colorado State University

Igor Steinmacher
Northern Arizona University

Anita Sarma
Oregon State University

Download This Presentation

doi.org/10.69924/20639.0301

[1] Co-First Author
[2] Presenter



This material is based upon work supported by the National Science Foundation under Grant Numbers 2347311 and 2303043.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Background & Motivation

84%

of developers use or plan to use GenAI tools

2025 Stack Overflow Survey

Is it paying off? Developer opinions vary...

“I no longer need to waste my time with repetitive tasks. There is always a way to automate with AI.”

-P369

“... the state of AI tools still produces very hit and miss outcomes and induces a ton of frustration.”

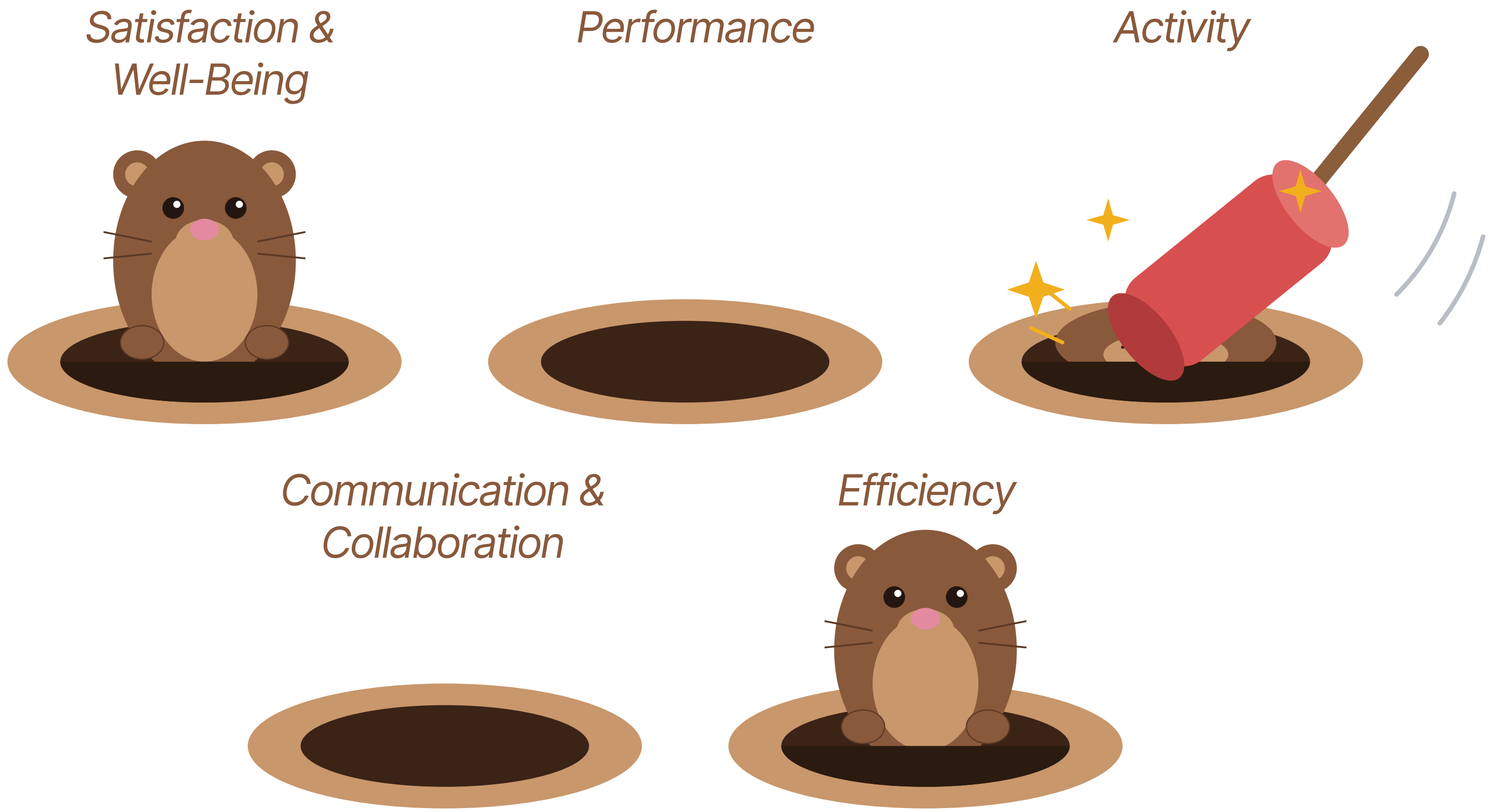
-P66

...and the evidence is mixed.

- ✓ Copilot users finished tasks up to 55.8% faster
  [Dohmke et al. 2025]
- ✓ AI cut time on a complex enterprise task by ~21% in a Google RCT
  [Paradis et al. 2025]
- ✗ Developers were 19% slower with AI
  [Tong 2025]
- ✗ Higher task-failure rates, no time savings
  [Vaithilingam et al. 2022]
- ✗ Only 29% trust the accuracy of AI output
  [2025 Stack Overflow Survey]
- ✗ 66% spend more time debugging than expected.
  [2025 Stack Overflow Survey]

What if we're measuring spurious productivity?

Does GenAI reduce developer effort, or just redistribute it to wherever the metrics are not looking?



Our analytical lens: measuring multi-dimensional productivity with SPACE.

Forsgren et al. (2021): productivity emerges from the interplay of human, technical, and organizational factors — not output volume alone.



Study Design / Methodology

Research Questions

RQ1

How does GenAI adoption affect developer productivity across multiple dimensions?

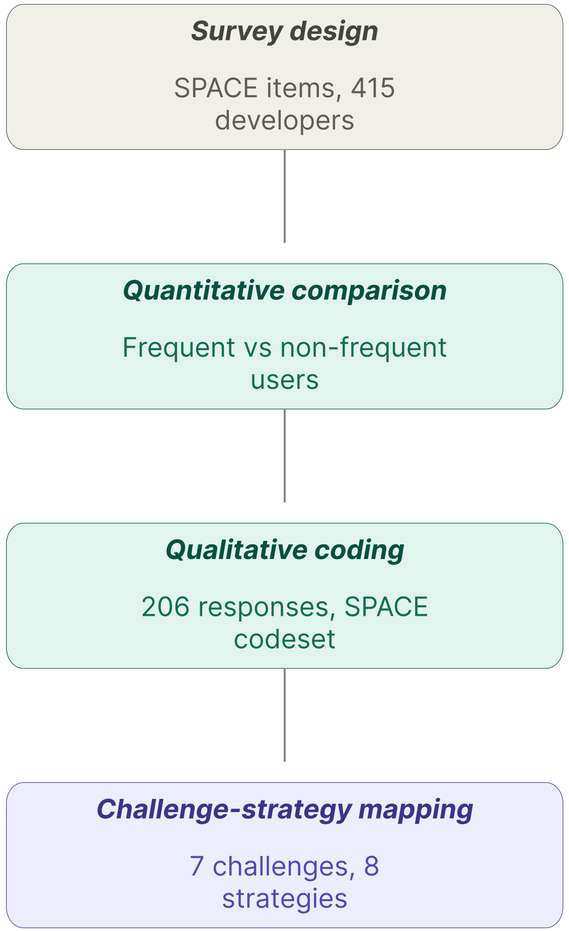
RQ2

What productivity-related gaps, challenges, and strategies do developers perceive in GenAI adoption?

Survey

- Survey items mapped to all five SPACE dimensions
- Split between frequent vs. infrequent users (violin plots, item-level charts with Likert distribution)

- Quantitative analysis of survey items
- Qualitative analysis of open-ended responses coded with SPACE (90% IRR)





Recruiting

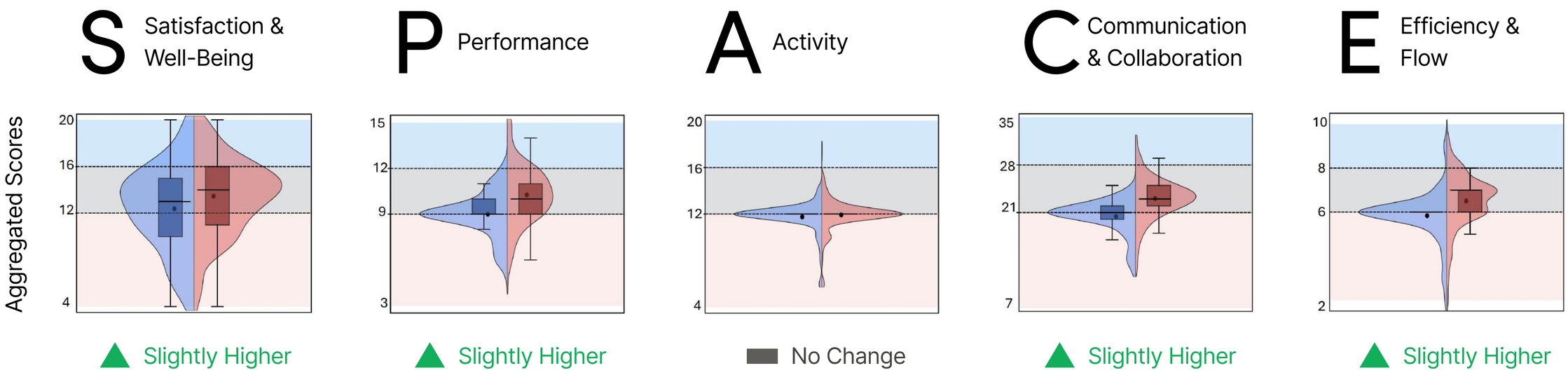
- Recruited by email & from 56 OSS communities
- 415 Professional Developers

- Mostly (90.6%) men
- Over half (57.8%) worked at large or extra-large organizations
- Most (82.2%) had more than five years of experience.
- Participants worked in:
  - Full-stack (36.9%)
  - Backend (16.9%)
  - Data/ML roles (15.4%)

Results

Aggregate view: barely a ripple

Median SPACE scores, frequent vs. non-frequent AI users are all within the neutral “no-change” band.



Satisfaction & Performance: gains with strings attached

Satisfaction

“I move fast with AI and move mountains of work, but I am losing my passion for the craft.”
-P201

- ✓ 68.6% of frequent users find workloads manageable (vs. 55.3%)
- ✓ 55.4% feel confident about job security
- ✗ 65.2% still report exhaustion — barely different from non-users (62.8%)

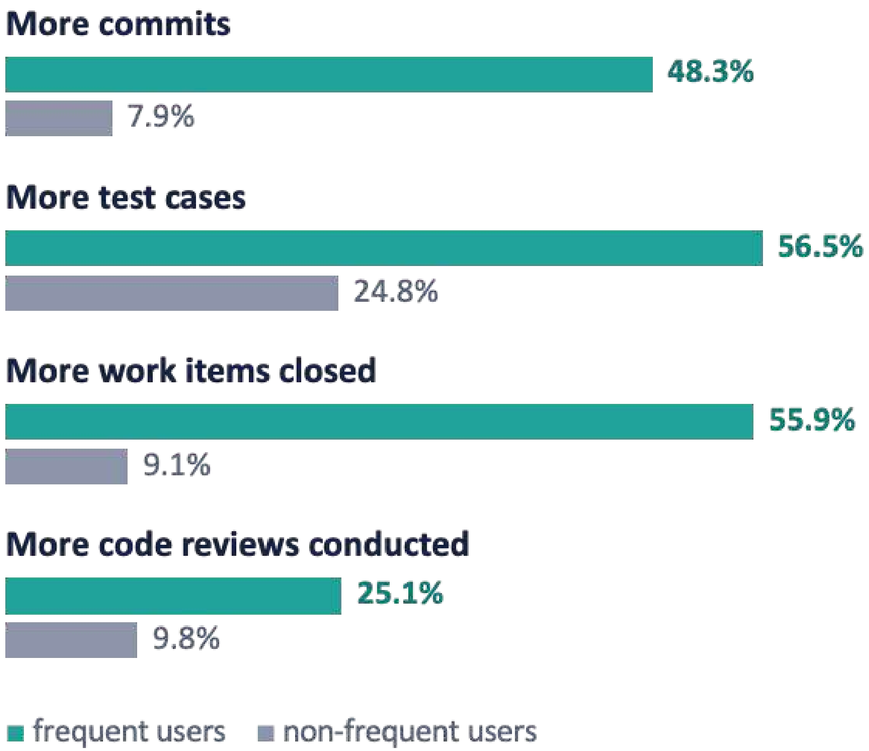
Performance

“LLM-based tools have so far been more about speed than quality improvement.”
- P5

- ✓ 72.7% of frequent users report higher coding throughput
- ✗ 67.4% see no change (or decline) in test pass rates
- ✗ 58.6% see no change in learning per day

Activity: more output, but more reviewing

Frequent users reported incrased activity...



...but there's a catch...

"LLM-based tools have so far been more about speed than quality improvement."
- P5

84.3% of frequent users say GenAI did not reduce time spent reviewing code

Collaboration untouched; team coordination still relies on human alignment

Collaboration



90%+ report no positive change in meeting frequency, meeting time, and emails written

Efficiency only partly improved

Efficiency

“I think AI can give you a productivity boost by bringing you 70% there in a few minutes. However, the last 30% will likely take you close to what you would’ve done before, because you also have to thoroughly review AI code.”
-P20

- ✓ Frequent users see less increase in time per work item (35.8% vs. 82.2%)
- ✓ Less personal browsing during work
- ✗ But 76.3% see no reduction in interruptions — flow stays fragile

Where does saved effort go?

"I'm able to close our features faster, but for the cost of significantly higher human brain compute load." -P207

Verification load

Cognitive workload of constantly checking, prompting, and fixing AI outputs (Ch1)

Review shift

Effort moves to reviewing others' AI-generated output instead of shrinking (Ch2)

Hype pressure

"AI = faster output" expectations from management raise stress (Ch3)

Verbose output

Wordy AI code and tests inflate review time (Ch4)

Skipped fundamentals

Leaning on AI before building foundational knowledge stalls learning (Ch5)

Individual-only support

GenAI assists solo tasks, not shared coordination & alignment (Ch6–7)

Strategies for Mitigating Effort Redistribution

- **Train, don't just encourage**
  Structured org-level GenAI training; pair AI use with fundamentals (St1, St5)
- **Frame AI as assistive**
  Team & management norms that resist "AI = more output" pressure (St2)
- **Surface Uncertainty**
  Confidence indicators + explanation norms (St3)
- **Give AI project context**
  Feed design decisions & architecture so tests reflect system intent (St4)
- **AI as first-pass reviewer**
  Use GenAI for triage, never final approval; quality gates needed for AI-heavy changes (St6, St7)
- **Norms for shared artifacts**
  Guidelines for GenAI in shared communication artifacts (St8)

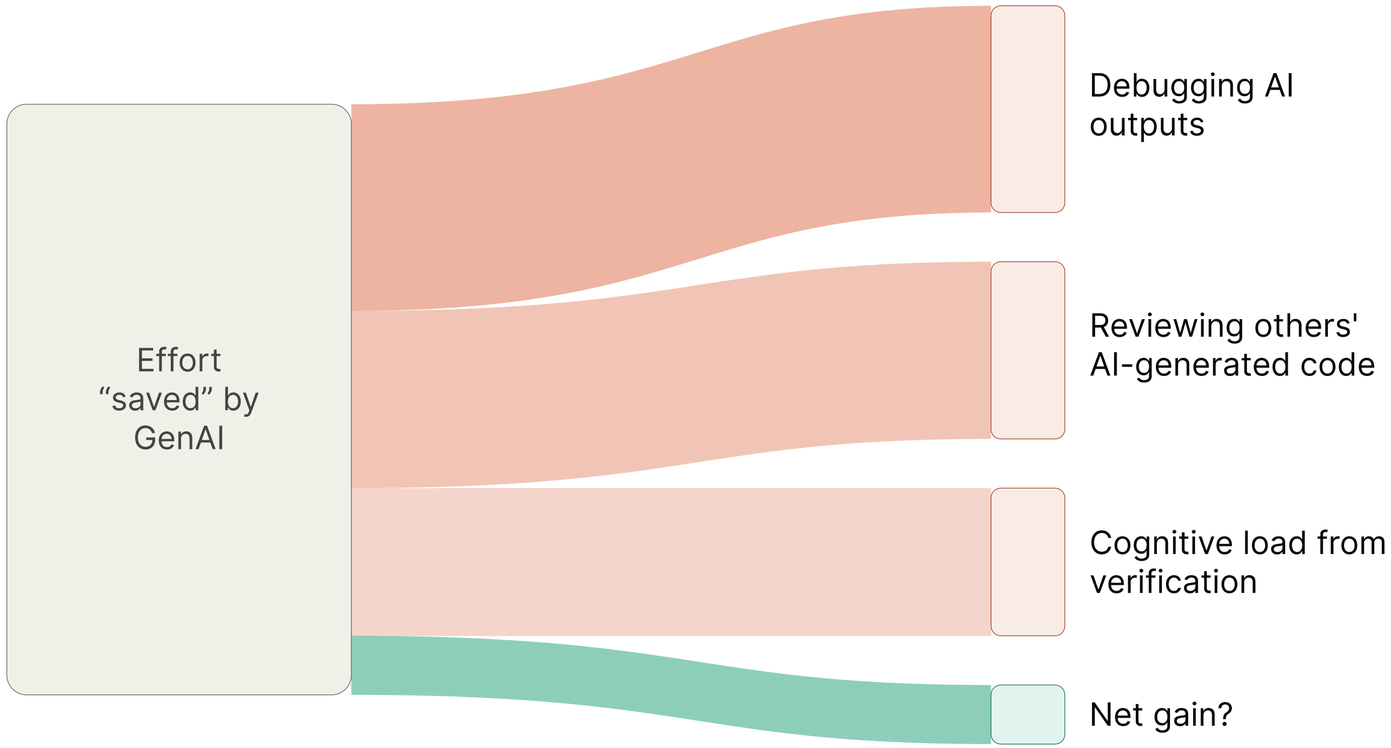
Takeaways

TAKEAWAY 1

Effort isn’t just eliminated, it may also be redistributed.

Companies and GenAI tool developers can use SPACE as a planning tool to predict where effort will shift before deployment.

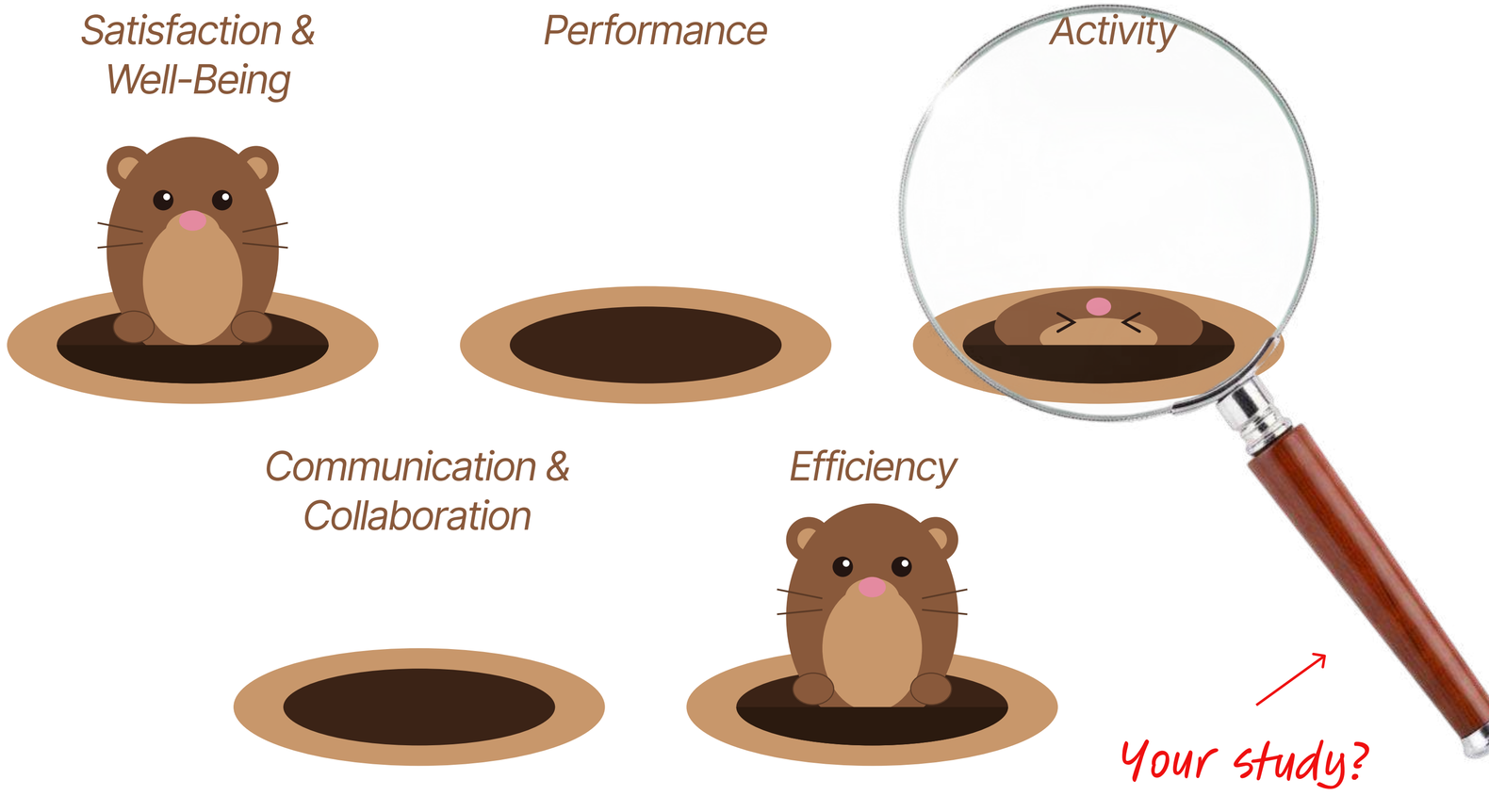
With careful constraints, can we prevent some redistribution?



TAKEAWAY 2

Studies must be careful to holistically measure productivity.

Measuring only volume mistakes motion for progress (cf. The Mythical Man-Month).



Download the Paper

doi.org/10.48550/arXiv.2510.24265

Contact Me

linkedin.com/in/tylermenezes
tylermenezes@codeday.org

References

[Dohmke et al. 2023] Dohmke, Thomas, Marco Iansiti, and Greg Richards. "Sea change in software development: Economic and productivity analysis of the ai-powered developer lifecycle." arXiv preprint arXiv:2306.15033 (2023).

[Paradis et al. 2025] Paradis, Elise, et al. "How much does AI impact development speed? An enterprise-based randomized controlled trial." 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE, 2025.

[Tong 2025] Anna Tong. 2025. AI slows down some experienced software developers, study finds. https://www.reuters.com/business/ai-slows-down-some-experienced- software- developers- study- finds- 2025- 07- 10/. Accessed: 2025-02-02.

[Vaithilingam et al. 2022] Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expec- tation vs. experience: Evaluating the usability of code generation tools powered by large language models. In CHI EA. ACM, New York, NY, USA, 1–7.