

3 WAYS TO **SABOTAGE** YOUR CODE WITH AI

**Analyzing 100m+ changed lines & 30m dev-years between
2022 and 2025: Where do good intentions go awry?**

Fight **Hype** with **Measurement**

Subjective “Data” About AI Impact

“I’d say maybe 30% of the code that is inside of our repos today is probably all written by software.” -Satya Nadella, CEO, Microsoft

“Computer programmers are now 10x more productive with AI assistance.” -Sam Altman, CEO, OpenAI

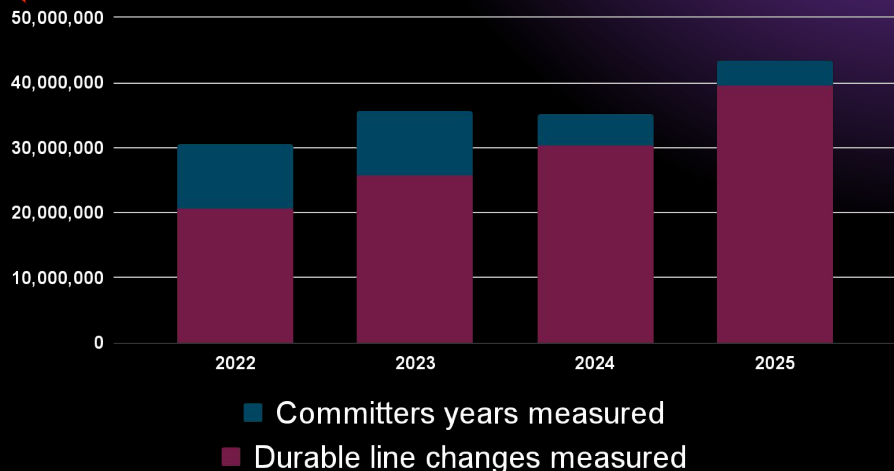
“AI could be writing 90% of software code within 3–6 months.” -Dario Amodei, CEO, Anthropic

At least they *tried* to quantify



Quantitative Data About AI Impact

Changes Analyzed by Year



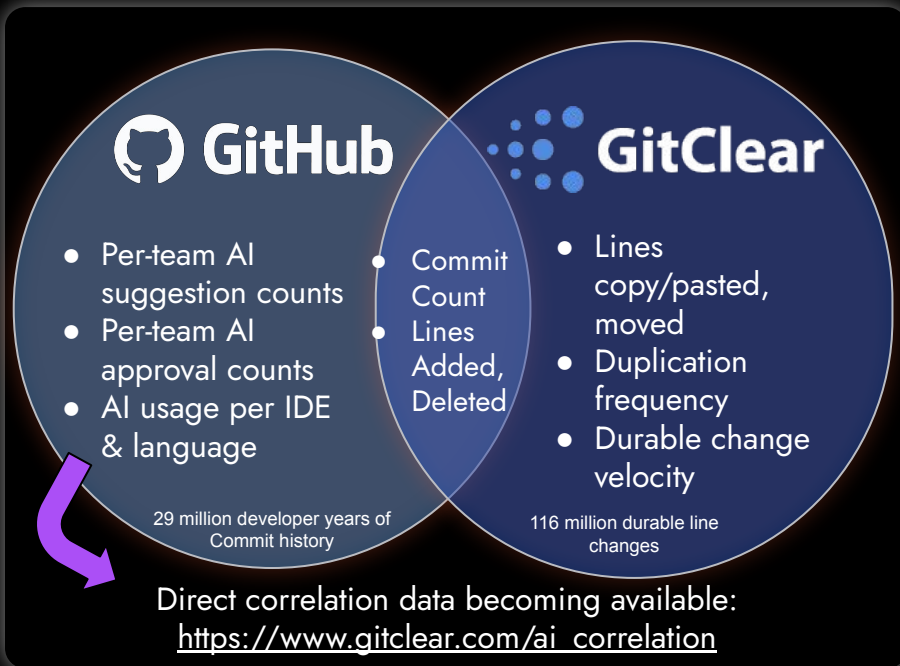
116 million durable line changes and **29 million developer-years** over 4 years, analyzed to help decision-making.
Data sources: Google, Meta, Amazon, Rappi, other small-to-midsize companies

AI Data Sources

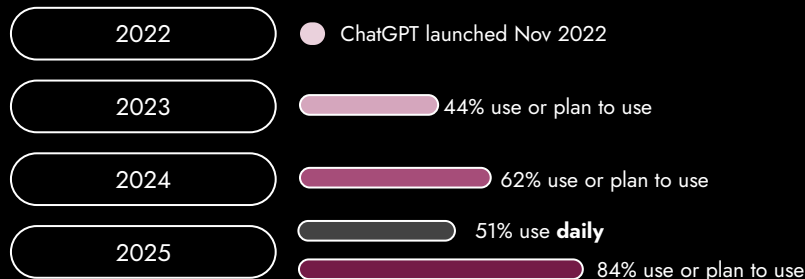
A hybrid of GitHub & GitClear data tables were used to discover multi-year trends that follow the rate of AI adoption.

Amount of AI Adoption by year is gathered from **Stack Overflow & JetBrains Survey results**.

New, directly measured, data from AI provider APIs is also being collected as of this quarter, creating new research possibilities.



AI Adoption by Year



Source: Stack Overflow Developer Surveys, 2023-2025

Quantifiable Correlation: Examples

What can we **quantify** about AI performance, quality & overall impact?



Velocity

As represented by **Commit Count**, **Diff Delta**, and **Issues Resolved**
Observed increases from 5-45%



Code Review Burden

Represented by **Review Count**, **Review Hours** and **Comment Count**
Observed increases from 10-20%



Code Quality & Durability

Represented by **Churn**, **Dupe Code Blocks**, and **Move/Refactor Activity**
Churn up by 50%, Duplication up by 80%, Refactor down by 50%

To reap the maximum reward, start by understanding where AI can be applied with **minimal developer pushback**.
I'll show how to create AI advocates **within** your organization.

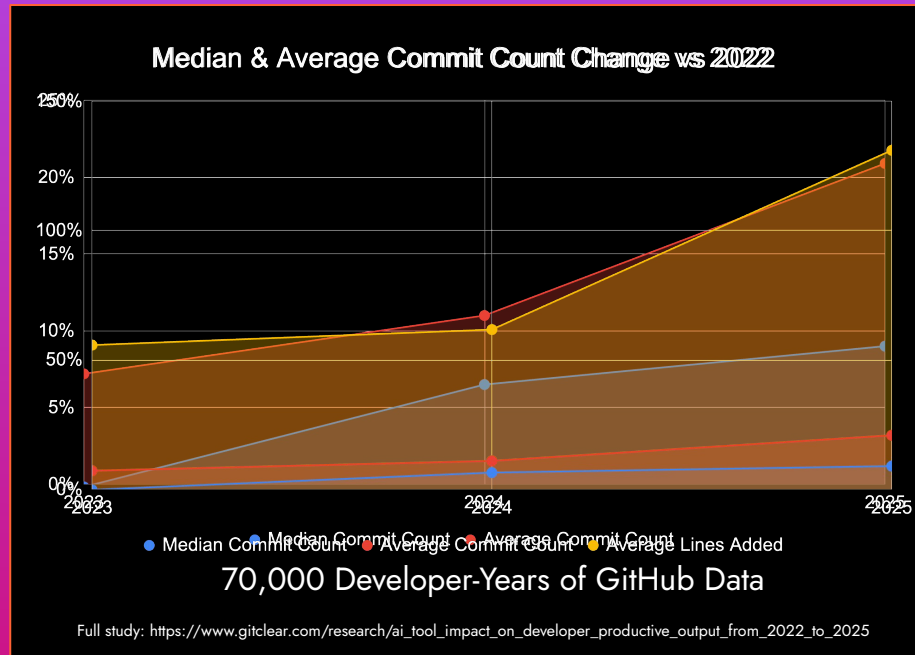
SABOTAGE #1 Sit out the revolution

2025's AI is **far from perfect**. But the biggest mistake is to sacrifice “good” to await “perfect.” Almost every relevant metric confirms at least a few percentage points of productivity gain available.

“But the most important metric, and we carefully measure it is, like, how much has our engineering velocity increased as a company due to AI, right? It’s tough measure, and we rigorously try to measure it, and our estimates are that number is now at 10%”

— Sundar Pichai, Google CEO, Lex Fridman Podcast

But what does it mean that the biggest spike is “lines added”?



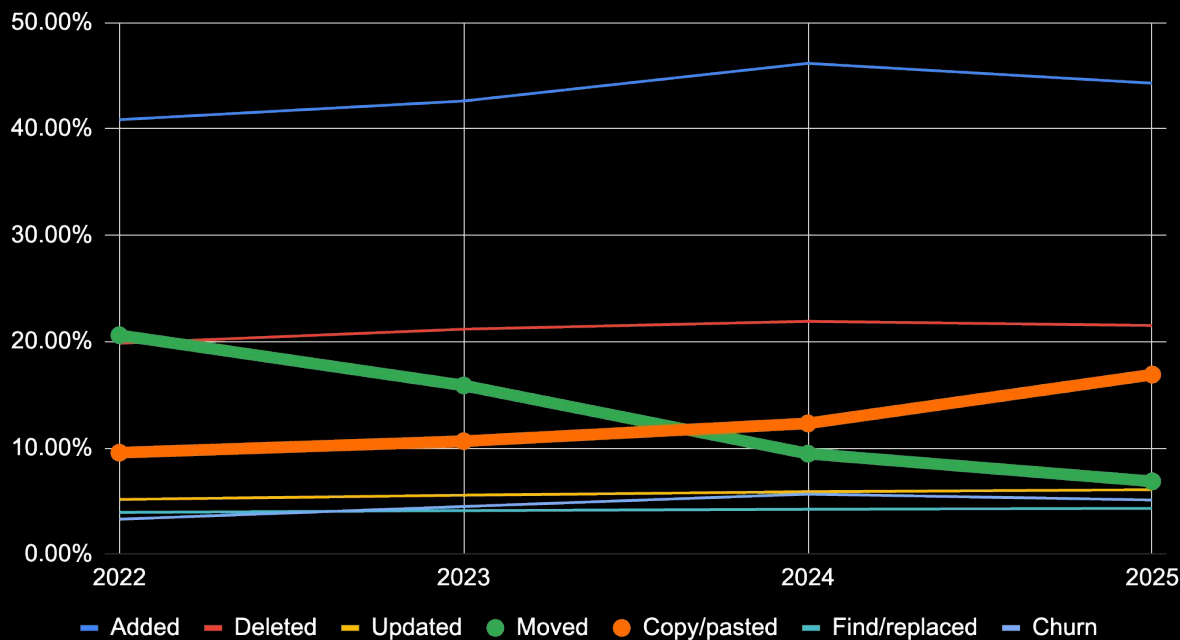
SABOTAGE #2

- A client was told by AI Sales that “other clients were seeing 2-5x gains.”
- And that’s why “what you measure” matters:

Change Type	2022	2025
Added line	40.9%	44.3%
Deleted line	19.9%	21.5%
Updated line	5.2%	6.1%
Move/refactor line	20.6%	6.9%
Copy/paste (dupe) line	9.6%	16.9%
Find/replace line	4.0%	4.3%
Churn	3.3%	5.1%

Mistaking “More Code” as “More Productivity”

Cost of generated code: **More copy/paste**, paired w/ **Less reuse**



Every line added is a contract to maintain code in perpetuity –
and the cost of maintaining copy/pasta is **high**

Is DRY Code Dead?

And if so, what are we losing?

As a Developer, is there any worse feeling than making swift progress on a ticket, then realizing there are two or three similar methods to choose from?

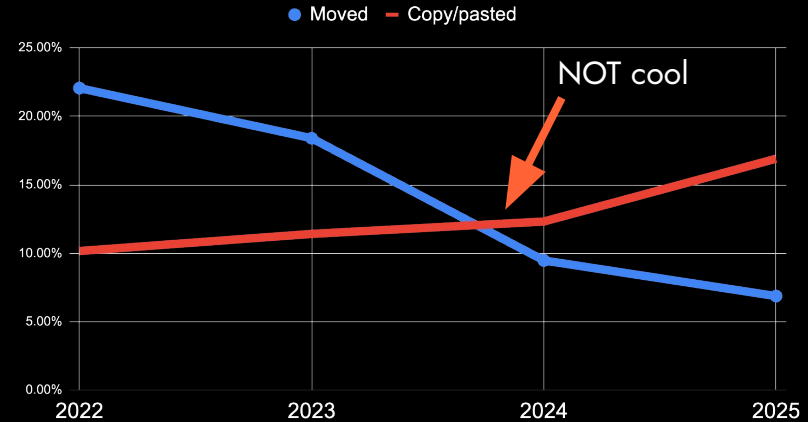
Questions Abound:

Which of the methods do you choose to use?

Why do more than one of these similar methods exist? Am I missing something?

Should I be a good citizen and consolidate or document these for future devs?

Refactor (Moved) vs Copy/Paste (Duplicate)



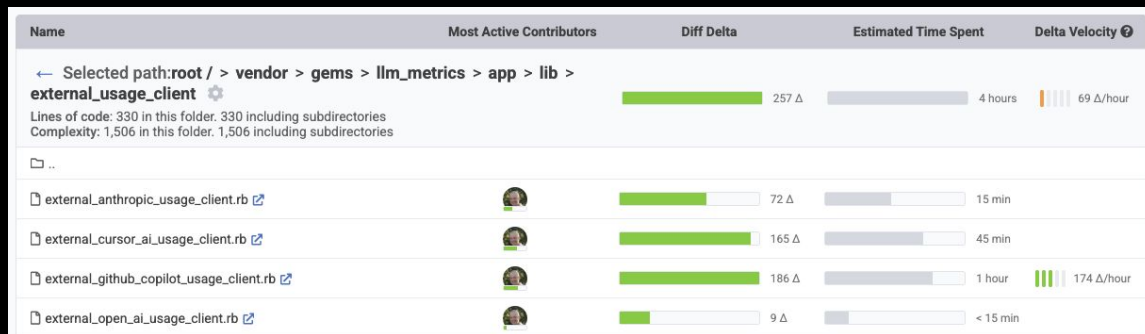
Sabotage #3 The “Willpower Drain” of Code Review

Case Study: Human-Coded vs Copilot Agent PR

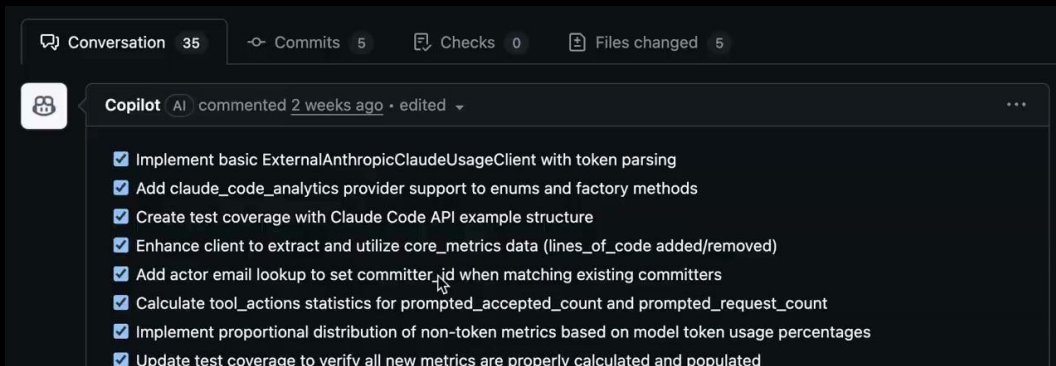
Pull Request Review Minutes

Year	Review Minutes
2022	33.5
2023	34.9
2024	40.7
2025	41.0

Source: GitClear Industry Stats



Human coded: In JetBrains IDE: 60 mins to develop, 30 mins to test API integration



AI Agent Coded: Over 4 review passes, 90 minutes to generate equivalent code

Proposed Heuristic

Evaluating a story? What are the numbers?

Proposed Heuristic

If someone is selling you a 3x gain from AI, it's probably a gain in tech debt.

Proposed Heuristic

If Google's CEO admits that the net AI productivity impact is on the order of ~10% in 2025... believe him.

Sidestepping **SABOTAGE**:
Heuristics to Consider

ACTIONABLE IDEAS TO APPLY

Developers loathe hype cycles. Win them over with **measurement**, **transparency**, and **successful examples**.

New AI API data released daily:

https://www.gitclear.com/ai_correlation

1. Acknowledge the Risks 🙄

Your own “move” and “copy/paste” data can prove that AI adoption remains a net benefit to long-term maintainability.

2. Prove the Benefits 🕵️

Developers are instinctively skeptical. Earn their respect with transparent data showing measurable AI upside.

3. Champion Top Cohorts 📢

Measure which team members have used AI best, and help them teach their tools & methods.

4. Set Realistic Goals 📈

Pick metrics that align with team success, and share data on how AI use is helping realize goals.