



# ***How Engineers Use GenAI - Turn by Turn, at Bank Scale***

An early look at bank-grade telemetry for GenAI-assisted engineering—what's instrumented now, what we're seeing, and what's next.

Matt Beane  
CEO, Cofounder, SkillBench  
Associate Professor, UC Santa Barbara

# ***Thank you, Gene Kim!***

- Came to the community with a study of LLM use in complex, professional tasks in progress
- Gene made introductions allowing for two follow-on studies, early peek today!
- One, qualitative, interview-based: on genAI in SWE/Devops
- One focused on “rocket science” (CTO, world-leading consumer goods): **turn-by-turn telemetry**

[← Back to blog](#)

# The death of the junior developer

**Steve Yegge** June 24, 2024

Warning: This blog post is somewhat speculative; the sky might not be falling. But my spidey-sense is definitely tingling. The way we are all doing our jobs in software is changing, potentially in big ways. So let's think of this as a thought exercise.

With that disclaimer, we're off!

I have been chatting with a bunch of both junior and senior developer-type folk at different companies lately. By cross-barfarticulating these conversations I've picked up on an emerging new pattern, which is that like 80% of you are way smarter than me. My goodness. What have I been doing?

But also another pattern, which is that a lot of people picked a bad year to be a junior developer. A whole lot of people. I wouldn't want to be just getting started in the industry today.

Not just the computer industry. Any industry. It's a bad year to be a junior anything.

My wife Linh and I are huge horror-movie buffs. So, to reflect the somber theme of today's rant, I will use horror movie titles to introduce our sections.

Let's start by talking about what's happening to junior developers.

# THE SKILL CODE

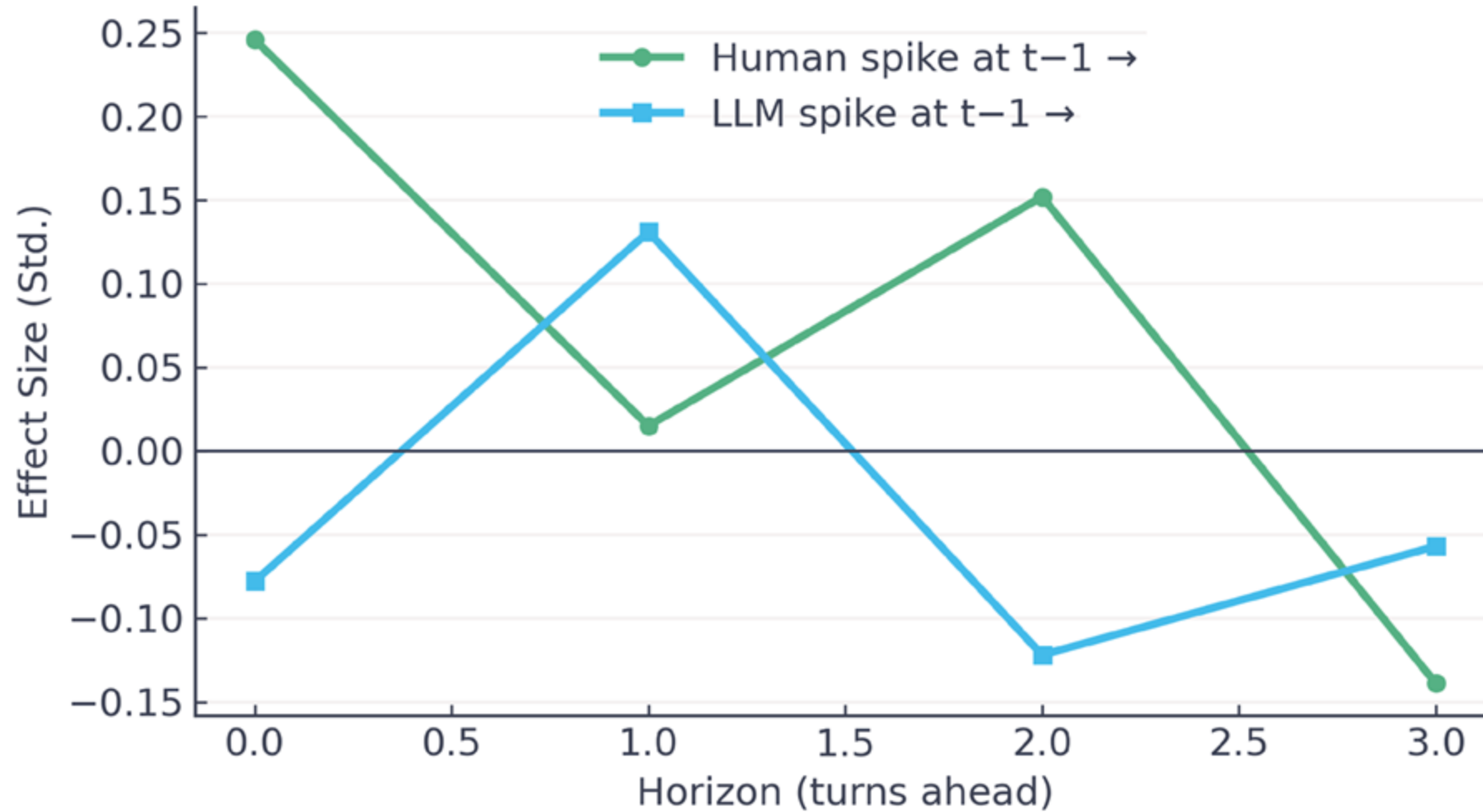


How to Save Human Ability  
in an Age of Intelligent Machines

**MATT BEANE**



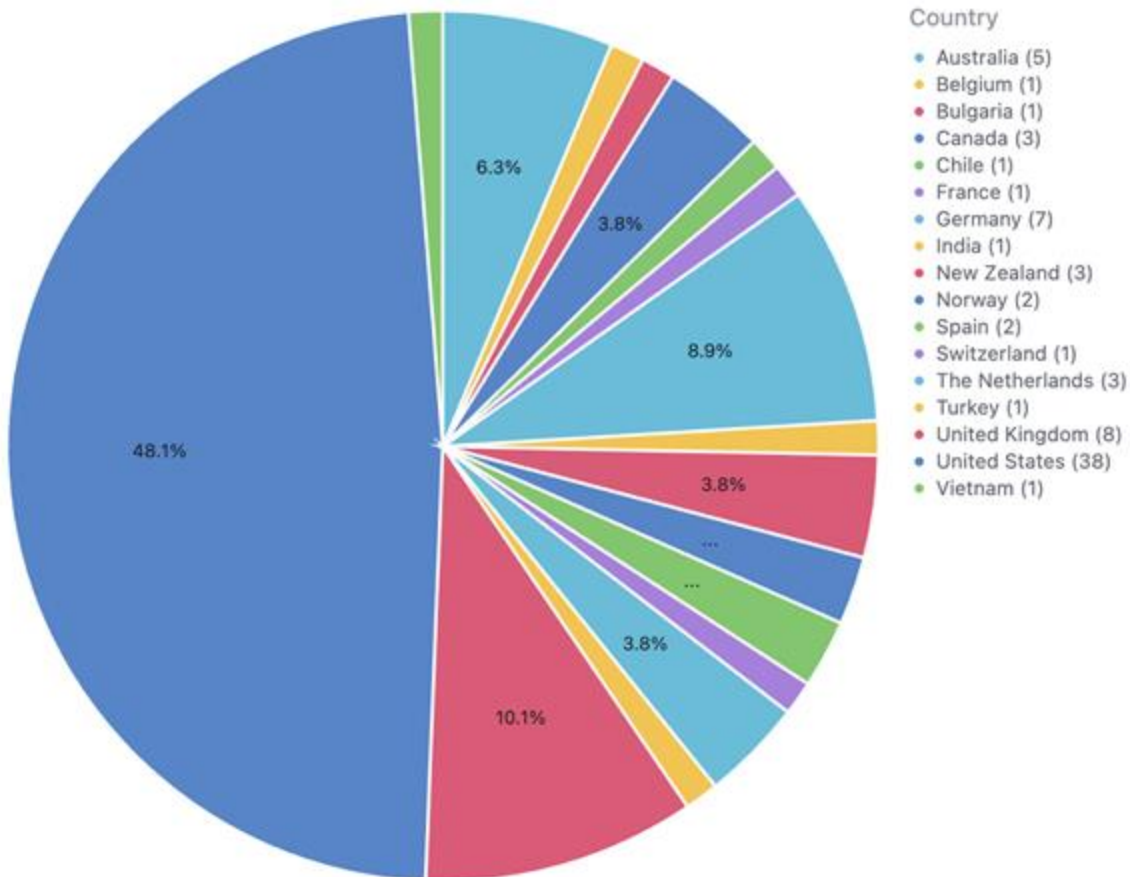
## Asymmetric Spillovers of [redacted]



# Interview Study: Diverse, Global Sample

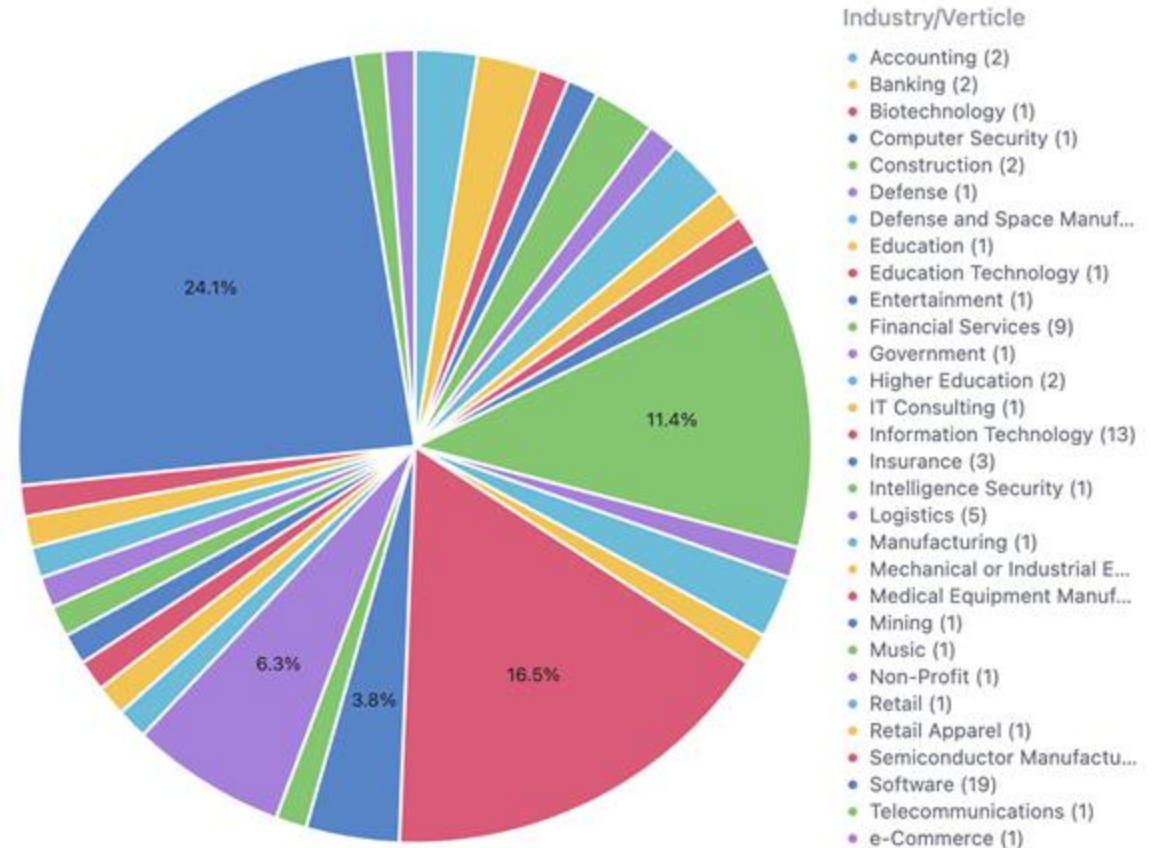
Percentage of Organizations by Country

Out of total number of organizations in any stage of the pipeline



Percentage of Organizations by Industry

Out of total number of organizations with Org Leader at Any Stage





# ***Finding 1: Boards, managers, devs want ROI***

**100%**

# ***Finding 2: ROI proof is a challenge***

## **Established Metrics**

### **Code Development**

- Breadth of usage: GenAI used in tasks
- Depth of usage
- Tokens per day
- Tokens per task

### **Code Review**

- AI code quality

### **Team Level**

- Team's breadth and depth of usage

### **Product Level**

- Product KPIs over AI features

## **Test/Ideal Metrics**

### **Code Development**

- Use quality: Story points/million tokens

### **AI-metered DevOps**

- Anomaly detection/resolution via log readers

### **AI Code Review**

- Syntax quality (Errors after review)
- Logic quality (Failures after review)

### **Team Level**

- Development Cycle Speed

### **Product Level**

- Portion of customer journey built via AI

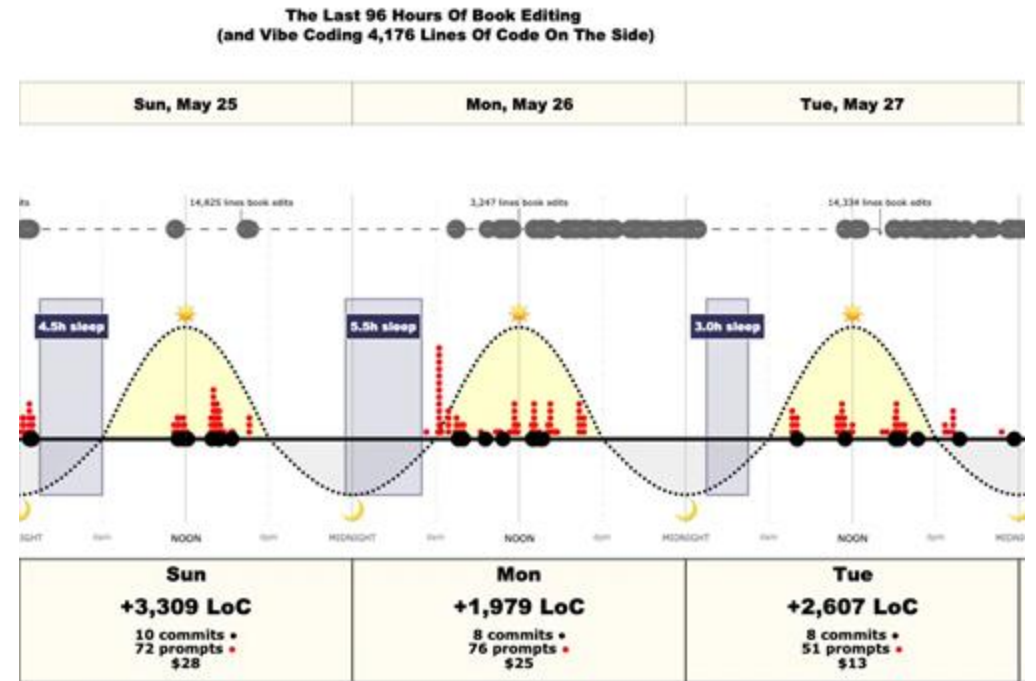
# Turn-by-Turn Telemetry: New Yardstick

ROI lives or dies in the sequence of small choices we make with AI

prompt → accept → edit → test → review

Turn-by-turn telemetry enables linking AI use to:

- quantifiable ROI
- productivity impact
- skill development



Gene's "Tufte Diagram" from the Vibe Coding book!

<https://itrevolution.com/articles/the-last-80-hours-of-editing-the-vibe-coding-book/>



# Answering the big Q with



Commonwealth  
Bank

How does variation in GenAI use change CBA engineers' work process and outcomes?

- Productivity, Quality, and Skill as target outcomes
- Turn-by-turn user interaction logs with GenAI
- ~120 SWEs opted in to our preregistered study
- Design is causal: two distinct "nudges" in AI use, four groups
- 2 weeks of data collected so far, will continue until Nov
- Preliminary, suggestive signals today



CIO for Technology,  
Commonwealth Bank

# CommBank Data Collection Overview

Collecting **real-time signals** of developer work from two sources:

- **VSCode Editor** - Capture actual code changes in real time
- **RooCode Agent** - Log interactions between developer and agent
- **GitHub PR** – Outcome signal



70599

Diff Logs



135

Chat Sessions



41

Pull Requests

# *Telemetry Signals from VS Code Diff Logs*

## Document Events

- **Document Open/Close/Save**
- **Text Changes** (with author attribution)
- **File Create/Delete/Rename**
- **Directory Create/Delete**

## Workspace Information

- **File Scan Summary**
- **File Content**

## GitHub Information

- **Repository Information**  
(owner/repository)
- **Branch Information**

# Secure Code Processing

Watch how we protect your code while preserving essential metadata.  
Your intellectual property remains secure throughout our processing pipeline.

main.rs

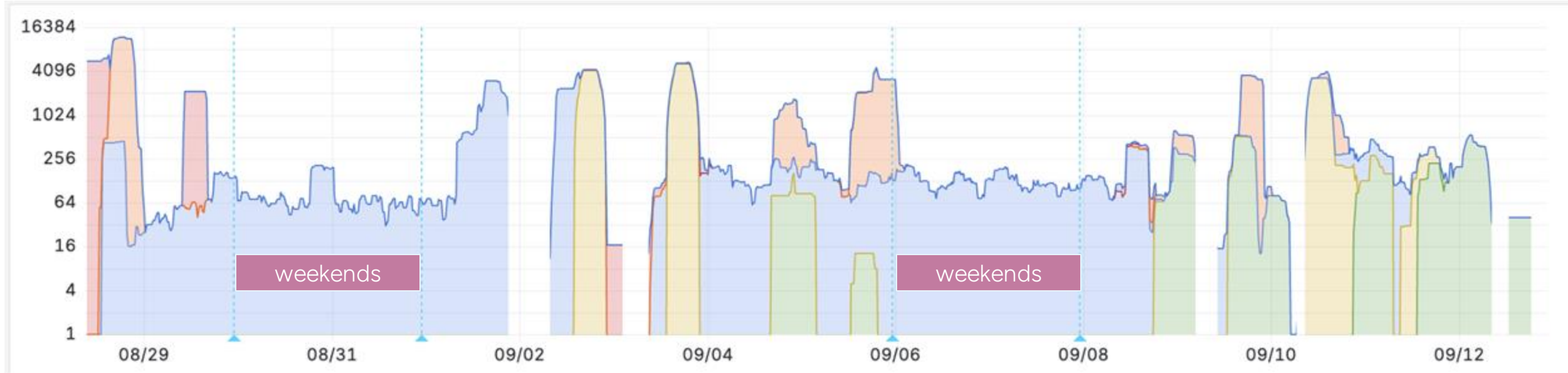
1

Start Coding Demo

Reset Demo

# Turn-by-Turn Activity Over Time

# of events



Each color represents a developer

# High-Level AI Interaction Summary

## AI Collaboration in Action

- **3,056 AI interactions** across **148 chat sessions**
- Average of **74 prompts per session**
- One developer used **732 prompts in a single session**

## Long, Iterative Workflows

- Average session length: **6.5 hours**

## Scale of Collaboration

- **72,763 code changes** across **14,131 files**

## Time Patterns

- Peak activity: **4 PM**
- Most productive day: **Thursday**

## AI is 10x-ing its Devs

- **Total characters:** 28,519,542  
Human: ~3.4M (□ 12%) vs **AI: ~25.1M (□ 88%)**
- **Total lines of code:** 916,573  
Human: ~72.7K (□ 8%) vs **AI: ~843.9K (□ 92%)**
- **AI tokens per session:** 14,697 (21x more than human)

## And Devs halfway accept their robot overlords

- **AI LOC accepted: 42.4%** (Nearly half of all code)



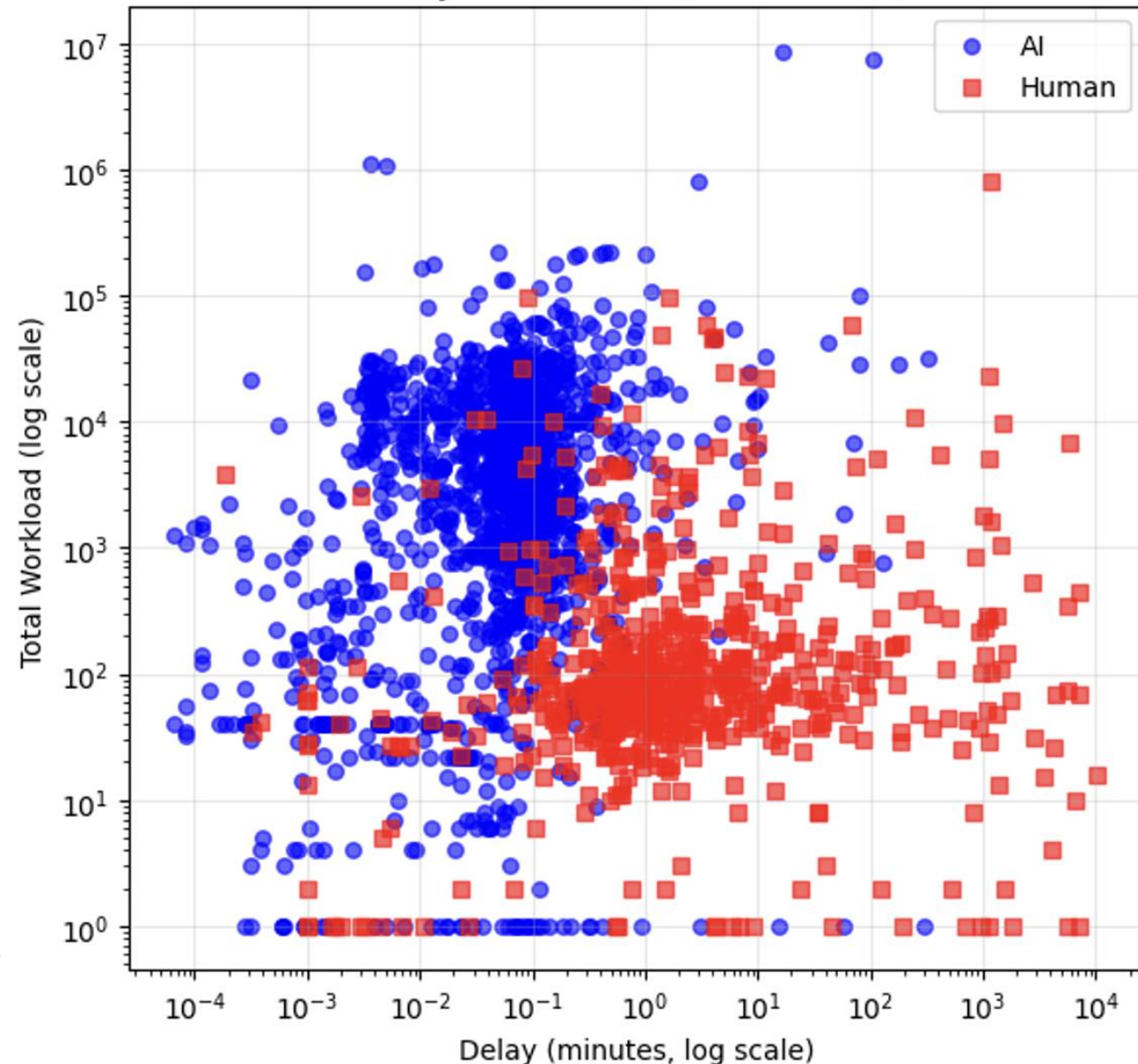
# AI vs. Dev Rhythm

**AI:** packs order-of-magnitude more work into short bursts, sprints with little downtime, predictable cadence

**Developers:** variable pauses, delay is not associated with task size, seconds-scale response appears to be copy-paste

**When is delay (in)effective? When is an instant avalanche of code (in)effective?**

Delay vs Workload (AI vs Human)



# Proposed metric: avg\_prompt1\_tokens

 SAY  Text

09:28:15

Give me the correct command to use inside the container for my project. the command example is "echo '{"timestamp":"2024-01-15 14:30:00"}' | kafka-console-producer --bootstrap-server localhost:9092 --<NAME> test-input"

Vs.

Only one in nine “superusers” regularly started with rich context and intent

 SAY  Text 00:20:48

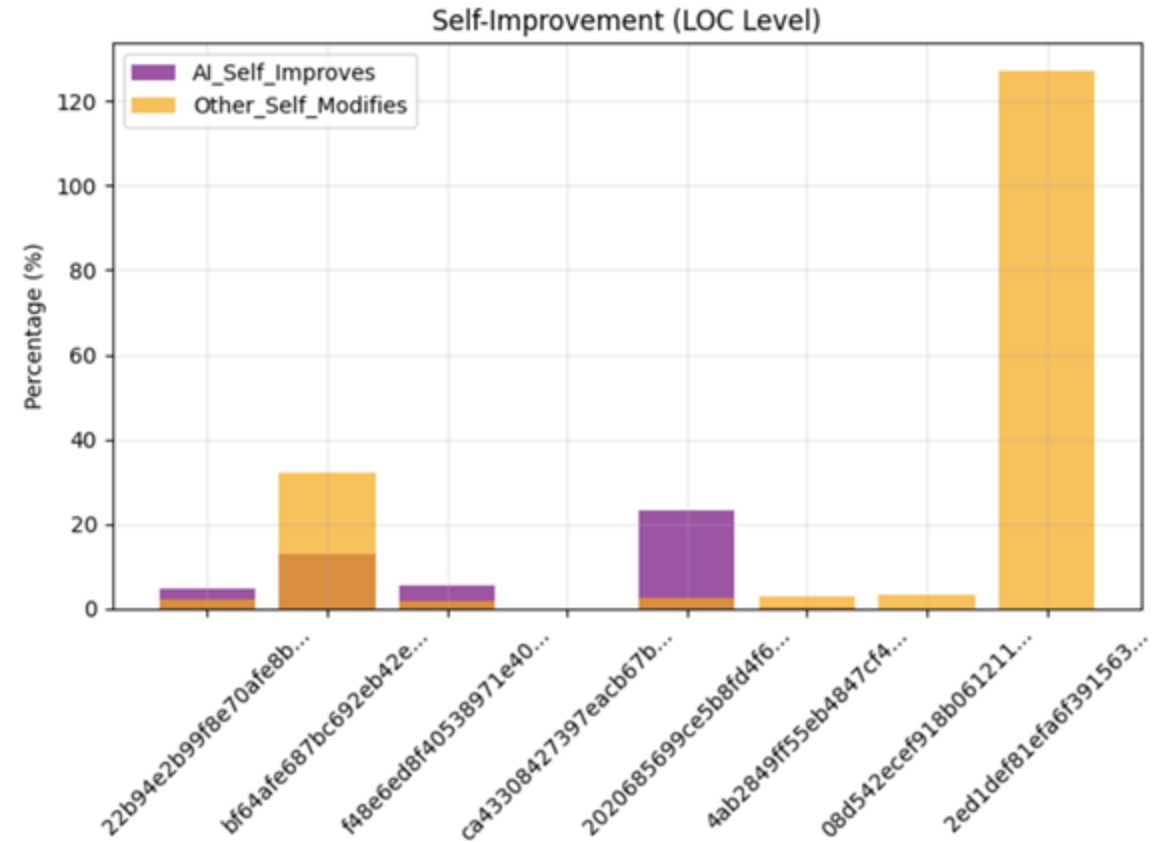
i belong to incubation 1 squad , below are the squad strategy , execution and milestone  
Incubation 1 - Strategy , Execution & Milestones Owned by <NAME> Last updated: about 7  
hours ago 1 <NAME> read 7 people viewed  Vision & Purpose To accelerate business  
transformation by incubating innovative workflow solutions, reusable assets, and  
integration patterns that improve time-to-market, resilience, and scalability of business  
processes.  Strategic Pillars Innovation Incubation <NAME> new ideas from business &  
tech stakeholders. Run quick PoCs on workflow tools (Appian, etc.). Explore AI-assisted  
workflow automation (NLP -> Process models, document -> workflow screens, etc.). Reusable  
Integration Patterns Build and standardize connectors/adapters ( Kafka etc). Create design  
patterns for common scenarios (e.g., event-driven orchestration). Acceleration Engine &  
experimentation Offer reusable accelerators (templates, <NAME>-built components, AI  
agents) PoC pipeline for emerging AI/automation capabilities (NLP, document ingestion)  
Rapid prototyping of new workflow features (e.g., low-code UI)  Business Alignment Align  
incubated ideas with top business challenges (e.g., Cards modernisation etc). Measure  
outcomes: time saved, reduced integration cost, automation uplift.  Ways of Working:  
Dual-track Agile: Ideation track (discovery) + Delivery track . Time-boxed incubation  
cycles (6-8 <NAME> per idea). Explore -> Experiment -> Accelerate -> Scale.  Milestone  
Model - Crawl, Walk, Run Milestone 1: Foundation & Incubation stage Enablement stage -  
Team enablement Build PoC's for reusable assets / components Milestone 2: Acceleration  
Scale successful PoCs into pilots with real business units use cases. Make reusable  
components available in market <NAME> Milestone 3: Adoption Extend the reusable components  
to other workflow tools/technologies (if technology specific) Measure impact (reduction in  
integration build time, process automation uplift). Engage community of practice across  
enterprise. Success Metrics % reuse of accelerators across business units. 2. Number of  
incubated - scaled ideas. 3. Measurable business impact (e.g., cost savings, SLA  
improvements). and for the next quarter we are planning to take below EPIC 2. Squad Epics  
Linkage to <NAME> OKRs : 1 - <NAME> Objective Top Down approach Owned by <NAME> 2 - <NAME>  
KR and Target Top Down approach Owned by <NAME> 4 - Squad Jira Epic contributing towards  
<NAME> KR's (include <NAME>) 5 - RID Please insert <NAME> to Jira if there is a RID for  
this Epic 1 - <NAME> Objective Top Down approach Owned by <NAME> 2 - <NAME> KR and Target  
Top Down approach Owned by <NAME> 4 - Squad Jira Epic contributing towards <NAME> KR's  
(include <NAME>) 5 - RID Please insert <NAME> to Jira if there is a RID for this Epic  
Objective : Incubate faster and reusable solutions through adoption of Strategic Workflows  
and Feature Marketplace KR2.1: Launch 3 new reusable workflow templates in the Feature  
Marketplace ?? Appian Component Reusability Assessment and Marketplace Enablement (DaaS  
and MSK) - Analyze existing Appian components developed by the IDP team to assess their  
reusability potential across the organization. Transform identified components into  
enterprise-<NAME> reusable assets and publish them to the internal Marketplace for  
organization-wide adoption. Dependency on IDP Objective : Incubate faster and reusable  
solutions through adoption of Strategic Workflows and Feature Marketplace KR2.1: Launch 3  
new reusable workflow templates in the Feature Marketplace (Extension to 01 component)  
Exception Handling Uplift - Enhance the existing Exception Handling component built  
implement structured logging standards to enable real-time log streaming to Observe for  
improved system observability and incident response. Dependency on Observability squad  
Objective : Incubate faster and reusable solutions through adoption of Strategic Workflows  
and Feature Marketplace KR2.1: Launch 3 new reusable workflow templates in the Feature  
Marketplace Streamlined real-time WIM (Workforce Information Management) data ingestion solution from  
CommSee that enables selective data streaming, allowing consumers to subscribe to only  
required data subsets instead of processing the entire WIM dataset. Innovation Incubation  
Appian AI Capabilities Analysis - Composer Optimization and Prototype Generation - Conduct  
analysis of Appian's AI-powered capabilities, specifically focusing on Appian Composer  
optimization, and AI-driven prototype/UI generation to maximize development efficiency and  
output quality. Innovation Incubation AI-Powered Pega to Appian Data Schema Translation  
Engine - Develop an AI-powered solution that can intelligently analyze Pega data schemas,  
understand their structure and relationships, and automatically propose optimized  
equivalent Appian data schemas to accelerate platform migration and reduce manual  
translation effort. Engineers across the Group continuously improve their <NAME> and  
impact by adopting modern engineering practices, tools, and learning pathways that enhance  
autonomy, quality, and velocity KR6.1: 100% of engineers complete modern engineering  
training CF & Appian Upskilling, AWS AI practitioner Training and Certification Give me  
innovative idea which i can implement using AI tool and appian system

# Proposed metric: overwrites (trust in AI?)

Overwrite tracking suggests **trust in AI**, **collaboration dynamics**, and even **productivity modes**

LOC-level **overwriting patterns**: AI more often changes human code than the reverse

- AI overwrites Human: 3.56%
- Human overwrites AI: 0.96%
- AI self-overwrites: 5.81%
- Human self-overwrites: 21.51%



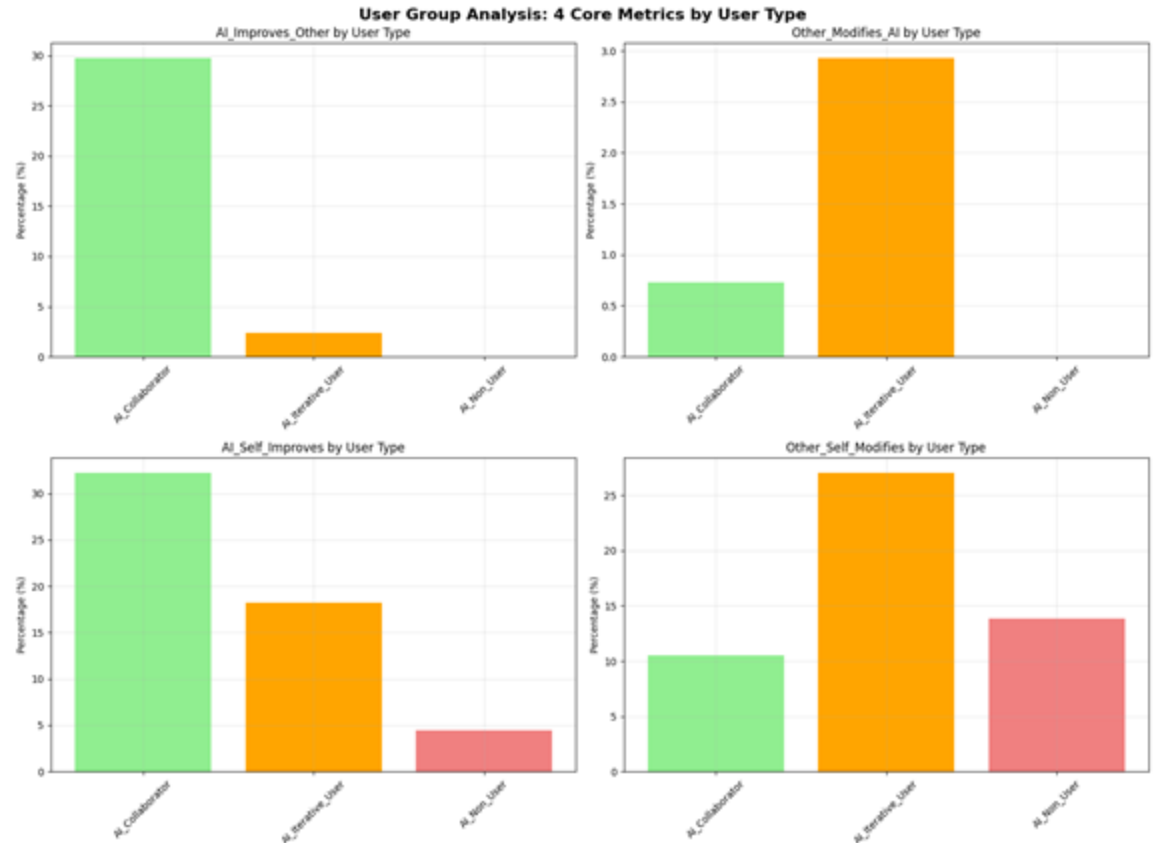
# And: trust varies by persona?

**Three data-driven personas of AI usage from overwrite tracking**

**Integrators:** High acceptance rate of AI code, AI improves human code, high trust

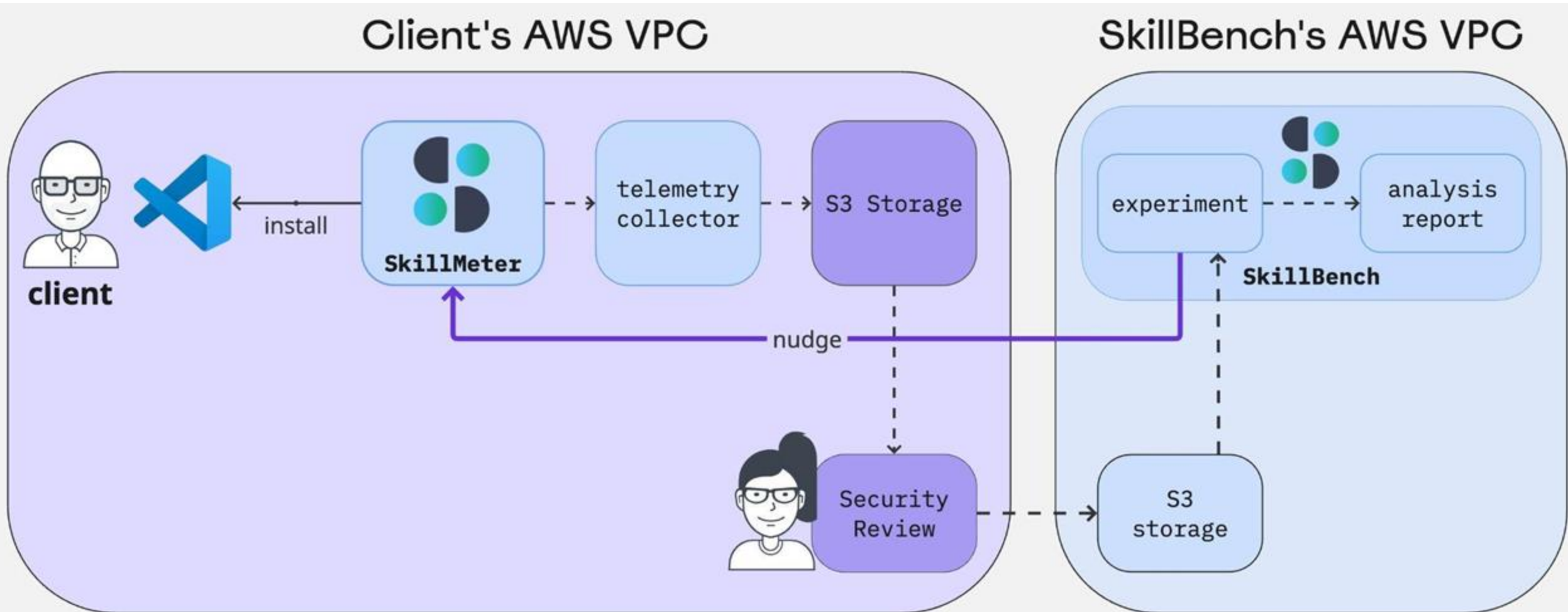
**Iterator:** Frequent refinements, human-driven iteration, AI as a sandbox

**Keyboardist:** Little/No AI usage





# Security-First Approach to Data Collection



# ***Early Lessons for Engineering Leaders***

- Measure sequences and traces, not just installs. Adoption dashboards miss much of the real story.
- AI and human work cadences differ. AI sprints nonstop; humans are more variable, pause often. Unclear when this is beneficial vs. not.
- Delay seems to be a deep/rich signal. Instant acceptance may mean copying; pauses may mean learning and synthesis.
- Avg\_prompt1\_tokens seems to matter quite a bit...
- Trust shows up in overwrites. Who edits whose code tells you about collaboration dynamics.



# ***Help needed: take our last pre-launch slot!***

We can onboard one additional enterprise partner in the next ~60 days.

**Interested?** Email [matt@skillbench.com](mailto:matt@skillbench.com) with subject: "ELTS - Telemetry"

**Ps: We're hiring!** If you or someone you know is interested in building with us, please reach out.