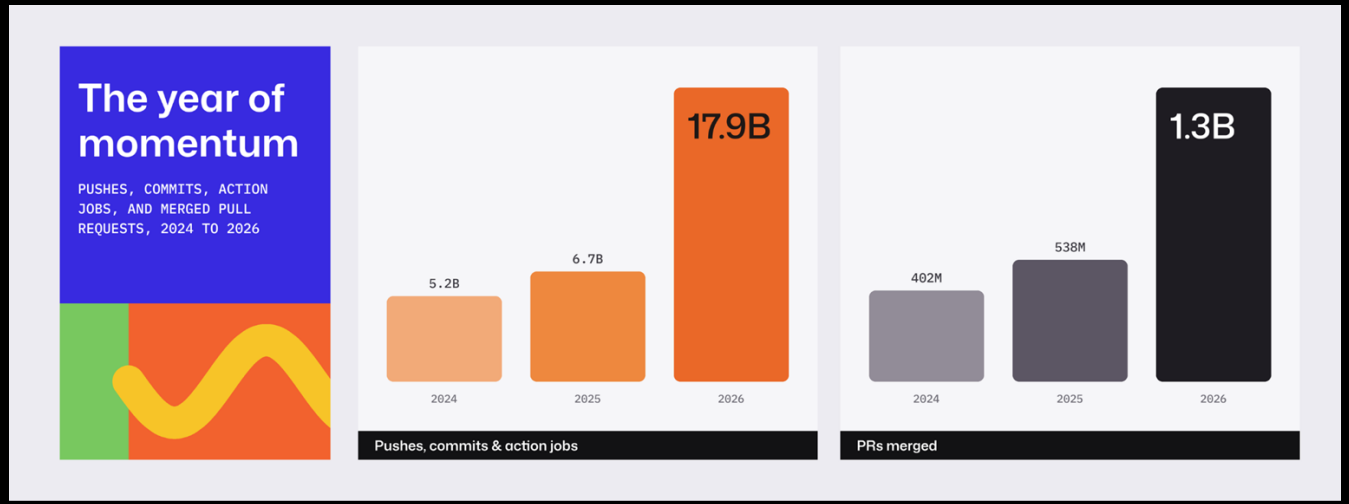
What successful engineering organizations do differently with AI

Roman Musatkin
Head of Product & Design @ Swarmia

/musatkin



Source: GitHub

Build the system that builds the product



swarmia

THE BUILD FRAMEWORK AT A GLANCE

1 AI LEVERAGE

Are we creating conditions for success for AI tools?

- AI adoption
- AI readiness
- AI autonomy
- AI development culture

2 EFFECTIVENESS

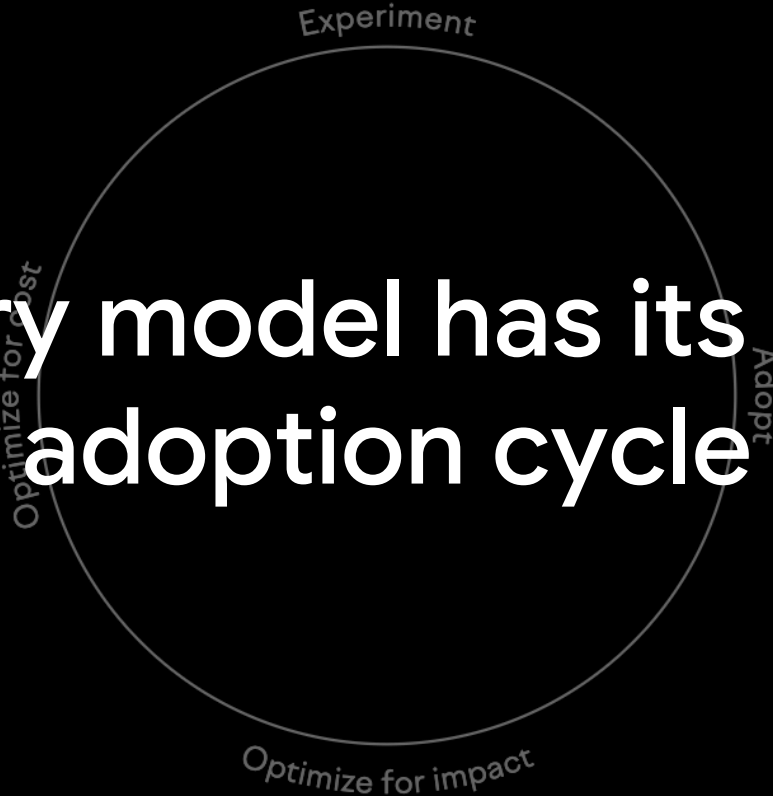
Are we delivering value faster, and is the pace sustainable?

- Developer productivity
- Quality and maintenance
- Developer experience

3 RETURN ON ENGINEERING

What business value does engineering investment produce?

- Business outcomes
- Engineering cost



Every model has its own adoption cycle

AI leverage / Readiness

Auto-improving AGENTS.md, skills in every repo

It’s easy to build and test prototypes with real data

Non-developers can do it, too

CI build time under 5 minutes

Testing environment for every change

It’s easy to release a feature flag

Custom agents in Slack / Teams for small fixes

UI/UX guidelines, tone of voice in code



AI leverage / Autonomy

Human-driven

Goal-driven

One prompt at a time

Auto-improving agent context

Triage and fix small bugs

Self-improving products

/goal build an end-to-end implementation of a new issue tracker

Self-improving documentation

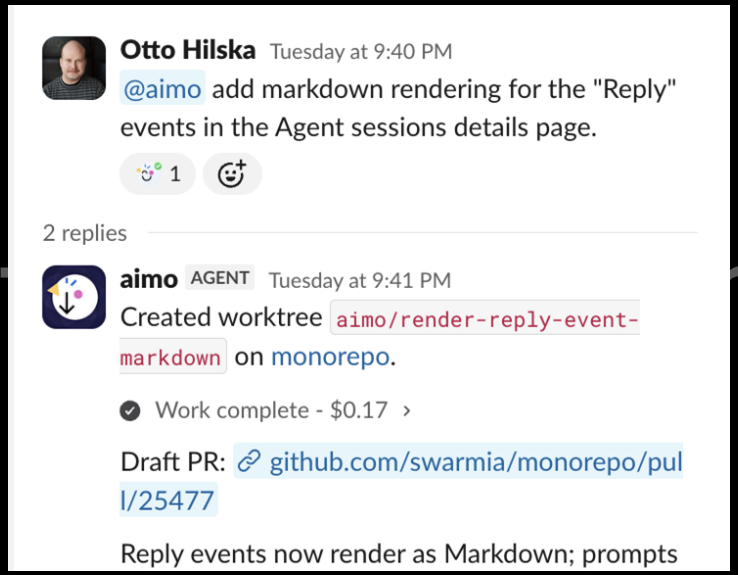
Auto-approve small changes



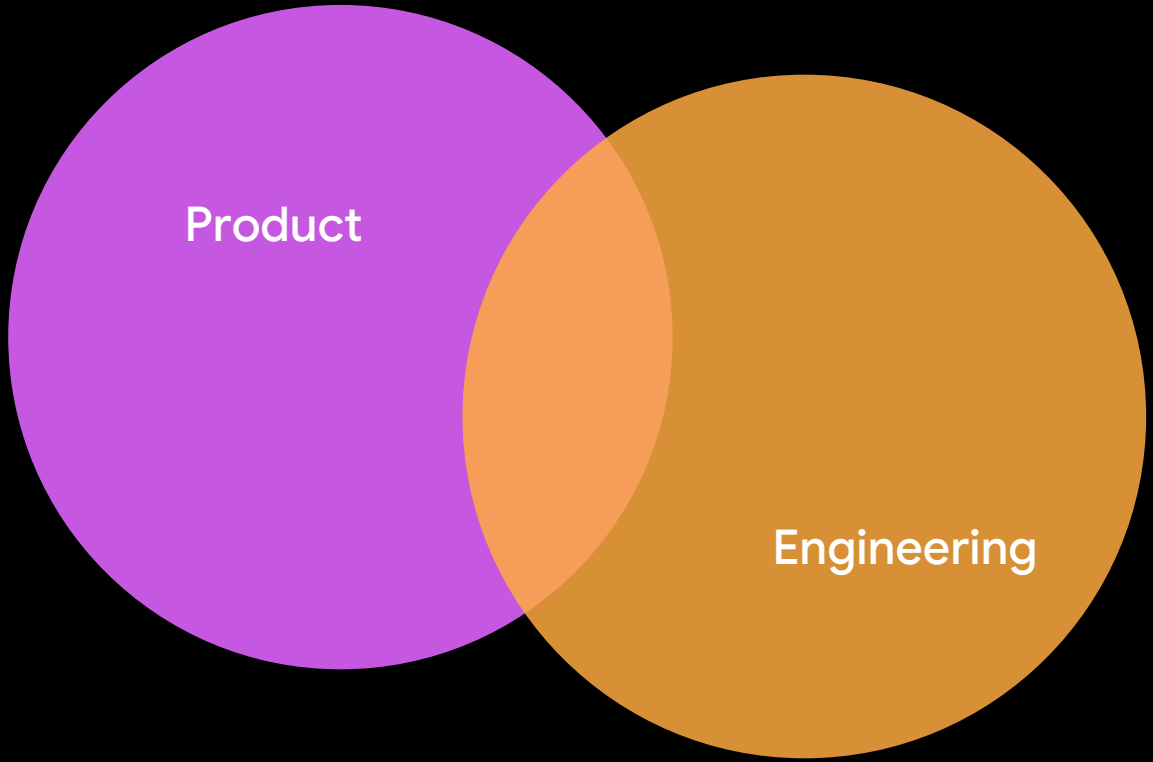
What was your prompt?



AI leverage / Culture

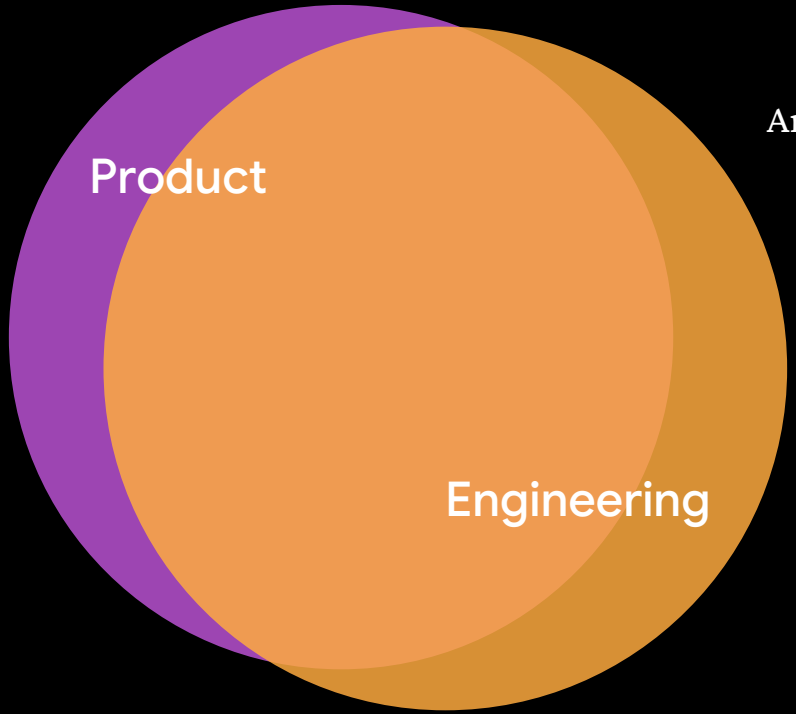


AI leverage / Culture



AI leverage / Culture

Speed of decision making, customer context, long-term vision



Architecture, systems, open-ended product problems

Build before you prioritize

2

EFFECTIVENESS

Are we delivering value faster, and is the pace sustainable?

DEVELOPER PRODUCTIVITY

- Throughput (combined measure of coding and issue throughput)
- Delivery (WIP, cycle time, batch size)
- Code review (time to first review, review load per engineer, AI-assisted review share, AI findings per PR)

QUALITY AND MAINTENANCE

- Share of new feature development vs. KTLO (maintenance, firefighting and rework)
- Bug backlog, number of incidents and change failure rate

DEVELOPER EXPERIENCE

- Developer satisfaction
- Cognitive load
- Organizational & system friction (ease of release, release confidence)

Speed and quality go hand in hand



Effectiveness / Quality and maintenance



1

Agents need better guidance to ship higher quality changes



2

Humans need better tools for dealing with the complexity.

Code review is not going away, *but it has to change*

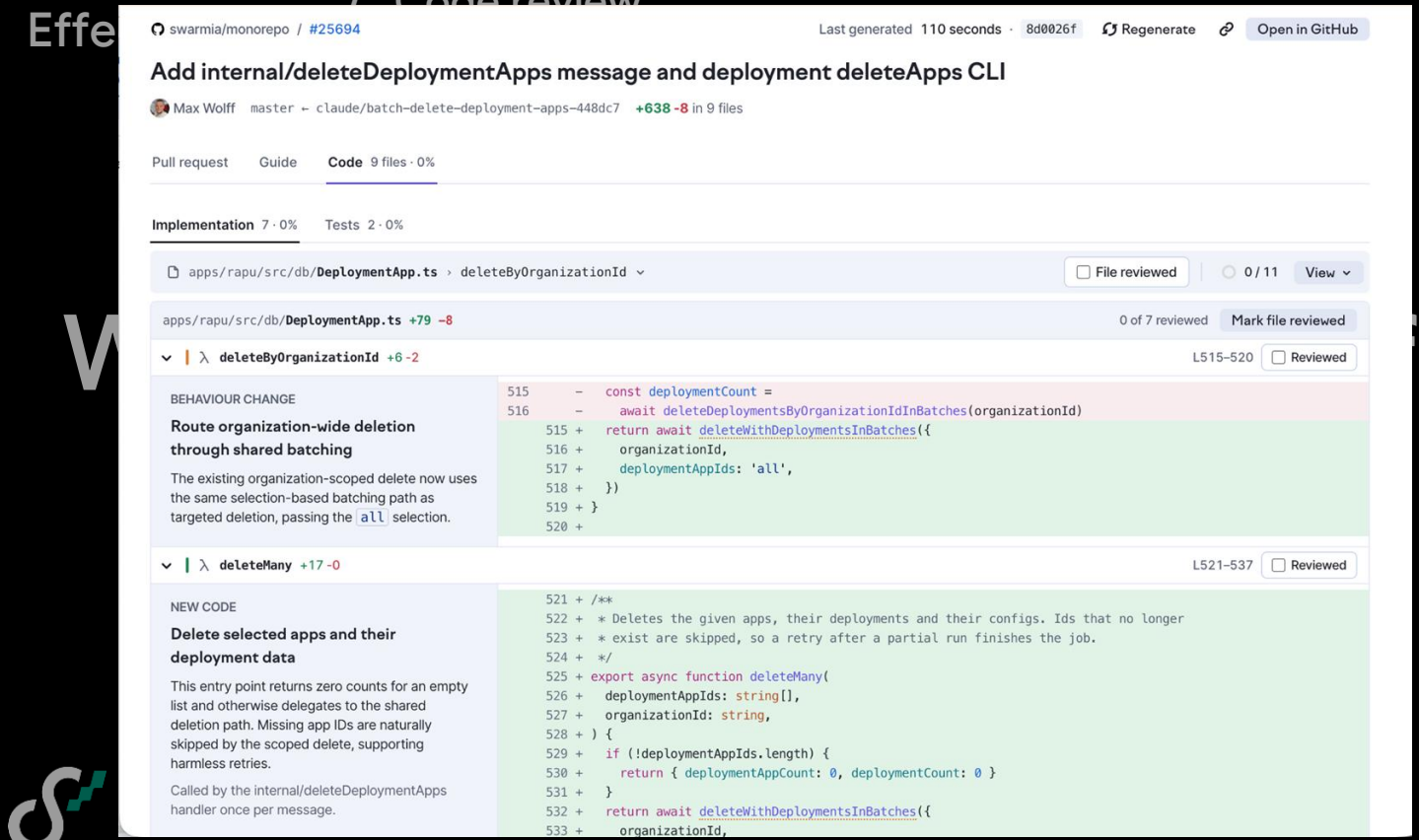


Why are we still viewing diffs in alphabetical order?



Effe / Code review

W S



Effectiveness / Developer experience

Are we starting more work than we're able to finish?

Do we use agents to ship more, or also maintain, and review, and fix bugs faster with better quality?

Is more of our engineering time drifting toward maintenance and rework?

Would we say the system is easier to work in than it was a year ago?





3

RETURN ON ENGINEERING

What business value does engineering investment produce?

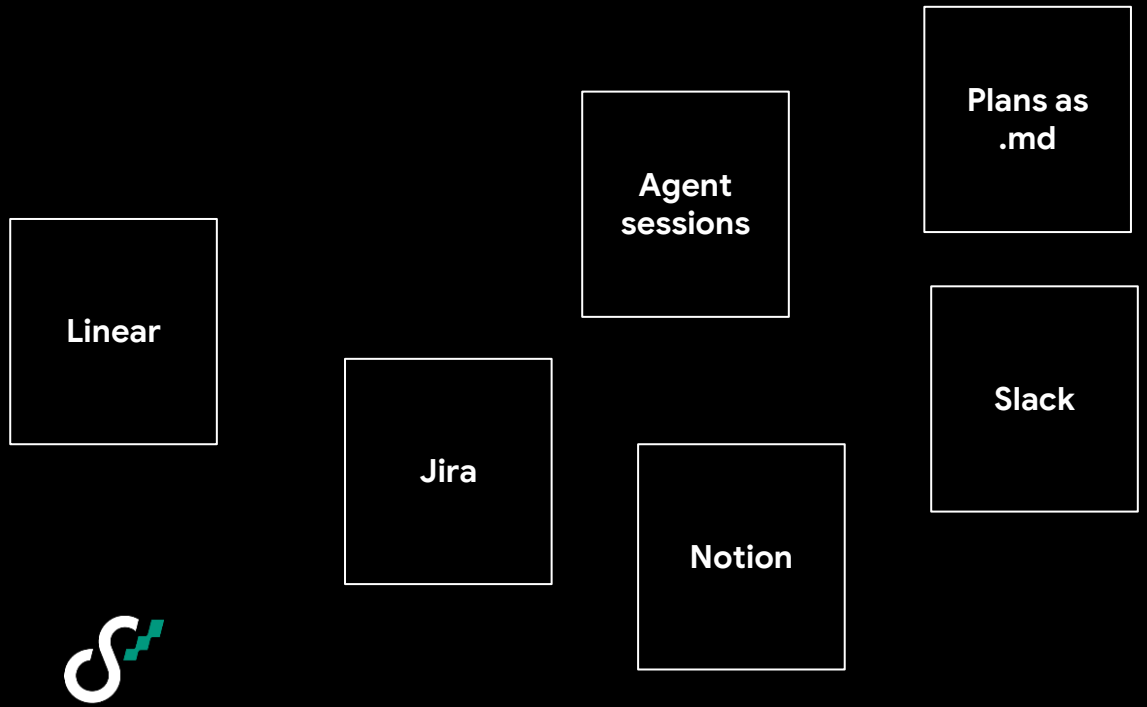
BUSINESS OUTCOMES

- Investment in new feature development and productivity improvements
- Share of work linked to organizational initiatives
- Progress towards organizational initiatives, initiative completion

ENGINEERING COST

- Combined engineering cost (headcount + AI)
- AI cost per FTE per month (total and share)
- Cost per unit of work (combined cost per user story, per pull request)

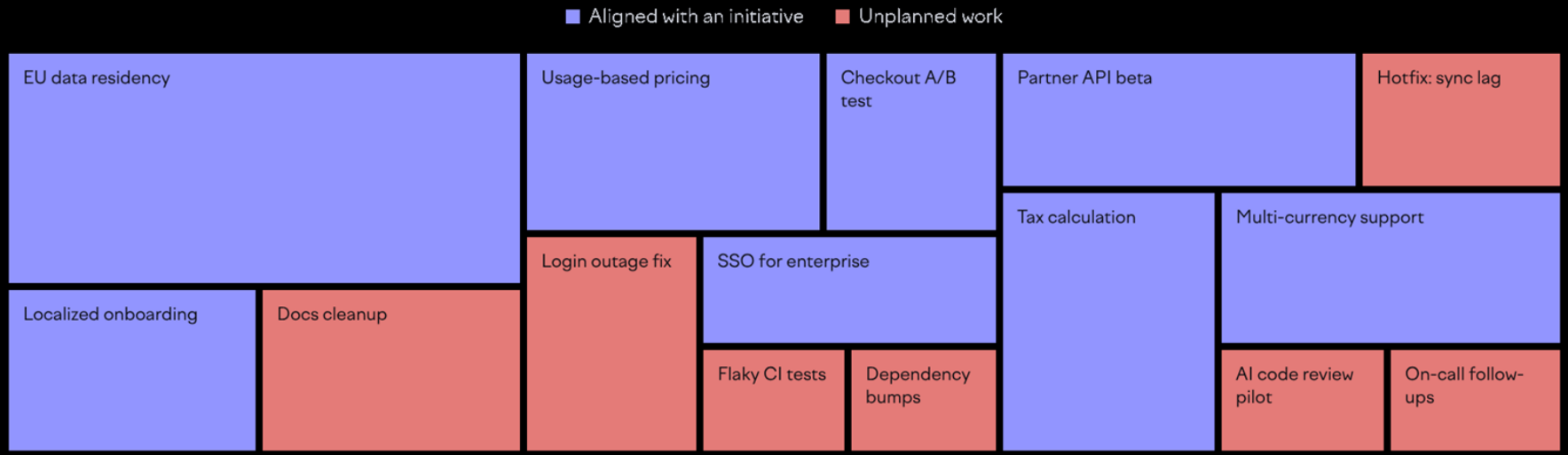
Return on engineering / Business outcomes



Less and less work is linked to a roadmap item

Factories are adding a new level of complexity

Return on engineering / Business outcomes



Return on engineering / Engineering cost

Last 3 months (incl. Oct) (vs. previous period)

AI token usage averaged $1,553 per engineer per month.

3 of 3 cost indicators trending in a positive direction.

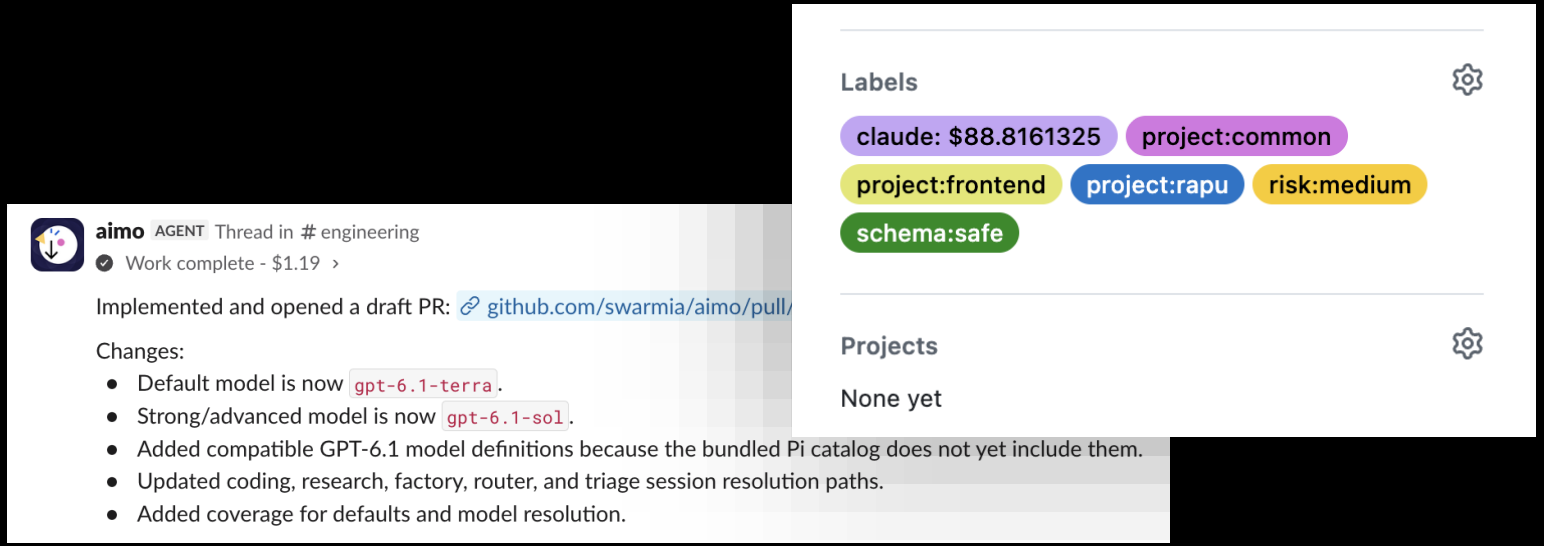
Total cost per story: **$11,550** -5%

Total cost per merged PR: **$236** -15%

Total cost per line changed: **$0.61** -19%



Return on engineering / Engineering cost



Build the system that builds the product

Let's talk about it.



swarmia.com/build

Roman Musatkin

/musatkin

The Build framework

swarmia.com/framework