opus

Agents on Call

How a lean team automated on-call with agents

Jeff Silver
Engineering, Opus

opus

Agents on Call

> How a lean team automated on-call with agents

OR how we accidentally built our software factory and coding app

Jeff Silver
Engineering, Opus

Opus in 60 seconds

ARR PER ENGINEER

$1.5M+

Growing fast, and profitable



Mobile-first training/ops platform for restaurants/service industry

Training 20,000 people per day across 60+ languages

~40 people

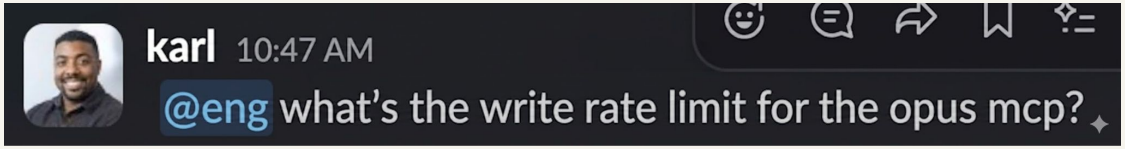
8 people on product team

A deliberately lean team

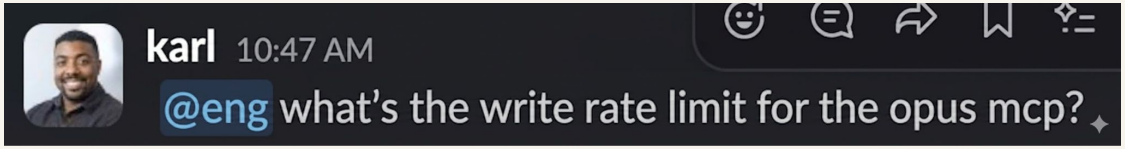
Leverage, not headcount. Maximize value each eng creates. Minimizing toil is key

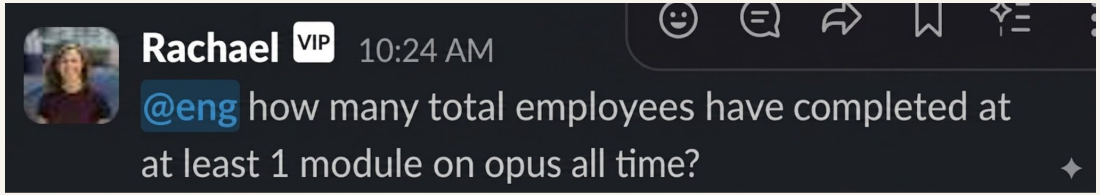
On-call became a nightmare

On-call became a nightmare

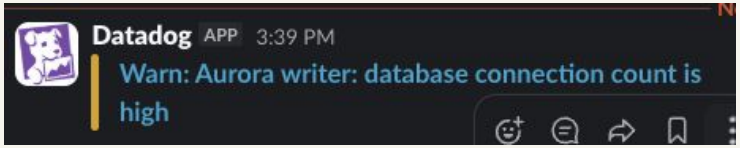


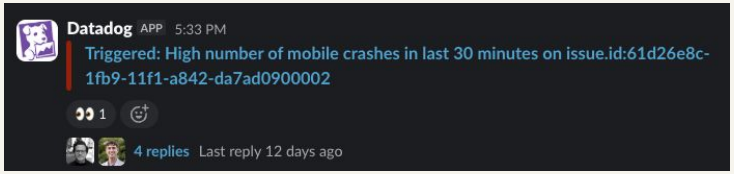
On-call became a nightmare

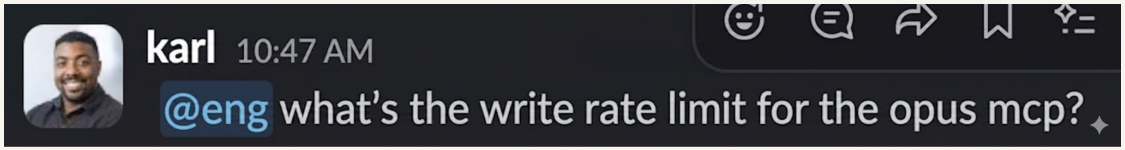


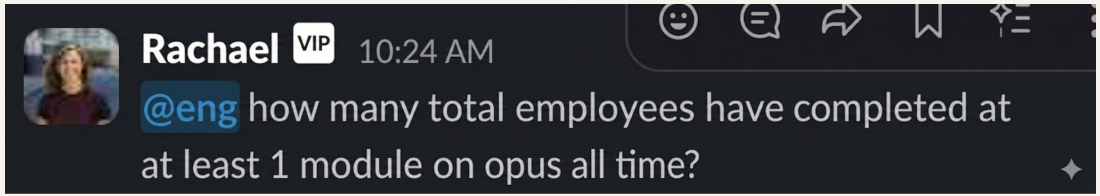


On-call became a nightmare

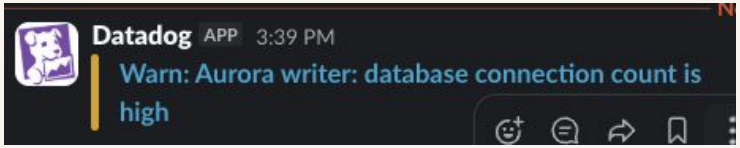


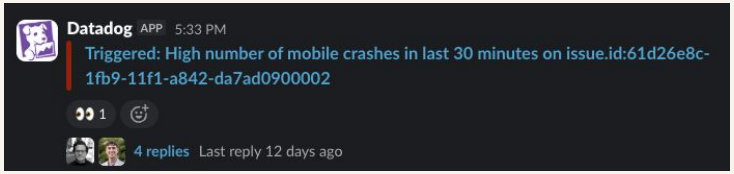


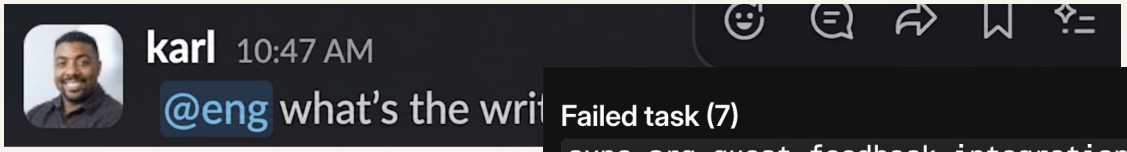


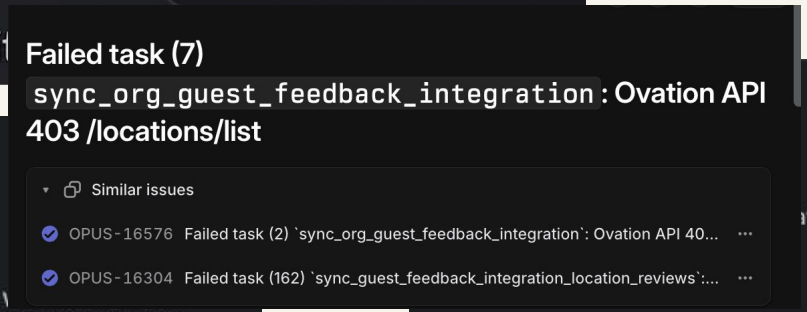


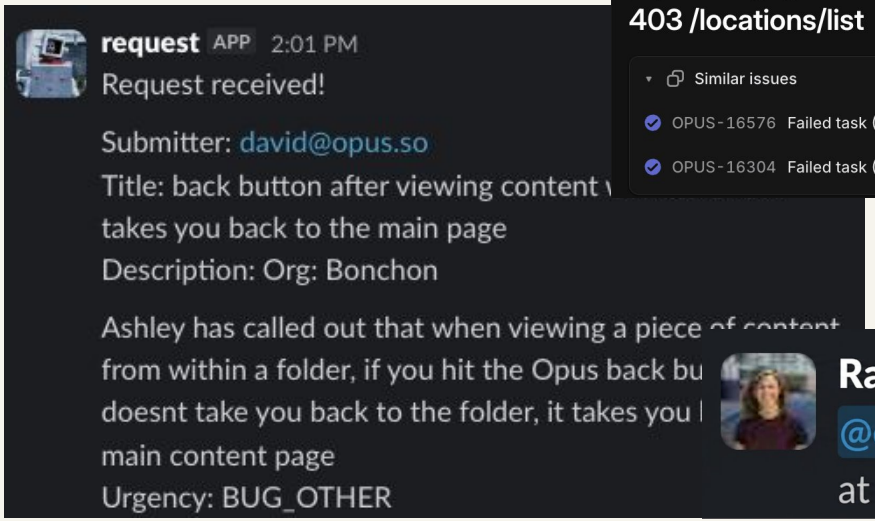
On-call became a nightmare

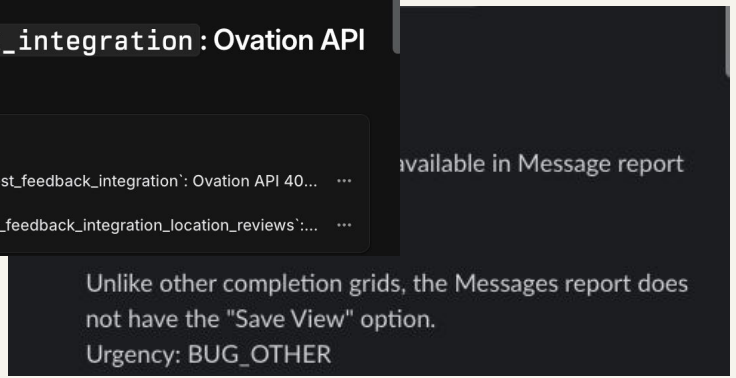


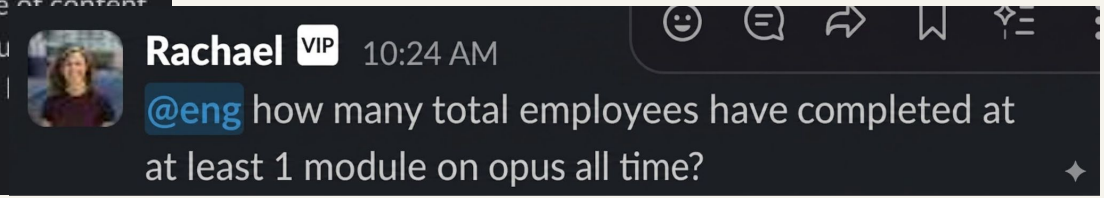








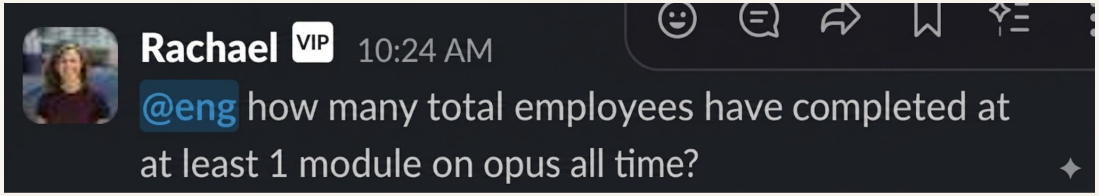


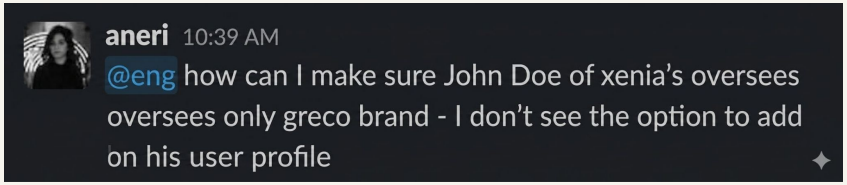


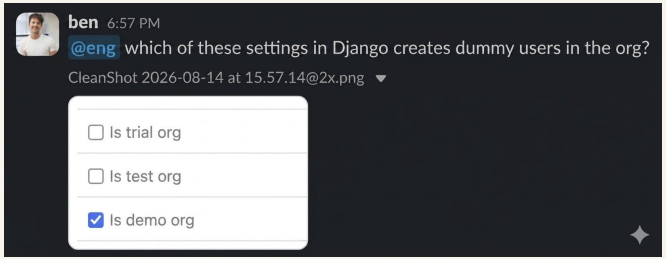
On-call became a nightmare

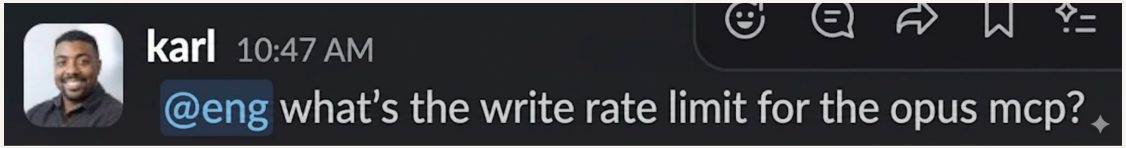


Start small....









CS/sales/ceo needs hyper-specific answers to unblock themselves

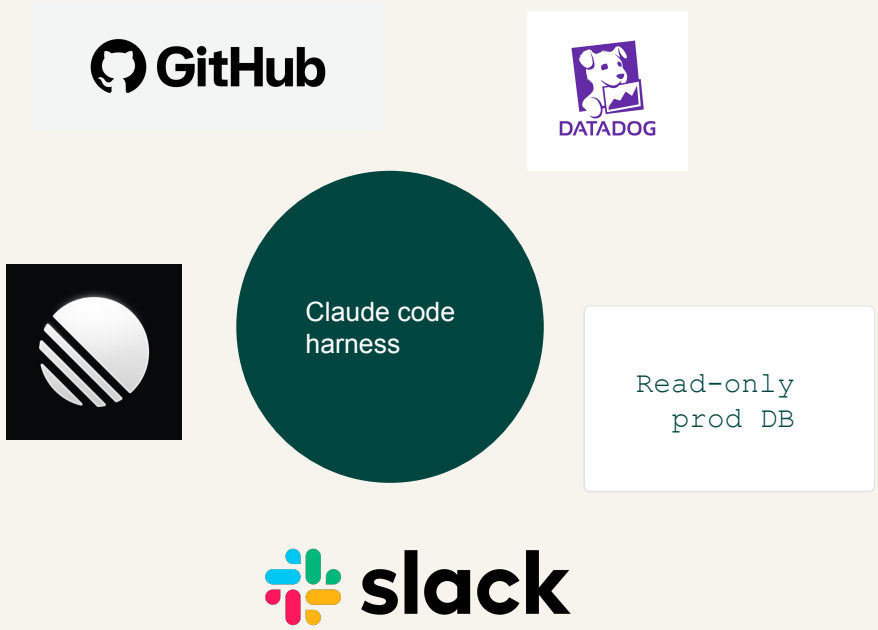
No runbook covers it, so every question goes to the on-call engineer (or me 😅)

We love helping our team mates, but this is not high leverage work for engineers to do

"🤔… Claude just got really good" - Me, November 2025

Unblock the company without interrupting eng

Unblock the company without interrupting eng



A Claude Code harness that runs in our own infra within our VPC

Invoked by a @mentions in slack, originally called Product Expert

Instant product market fit - CS demanded a rename

“Product Bestie”





Read-only
prod DB

A Claude Code harness that runs in our own infra within our VPC

Invoked by a @mentions in slack, originally called Product Expert

November 2025

Woah...

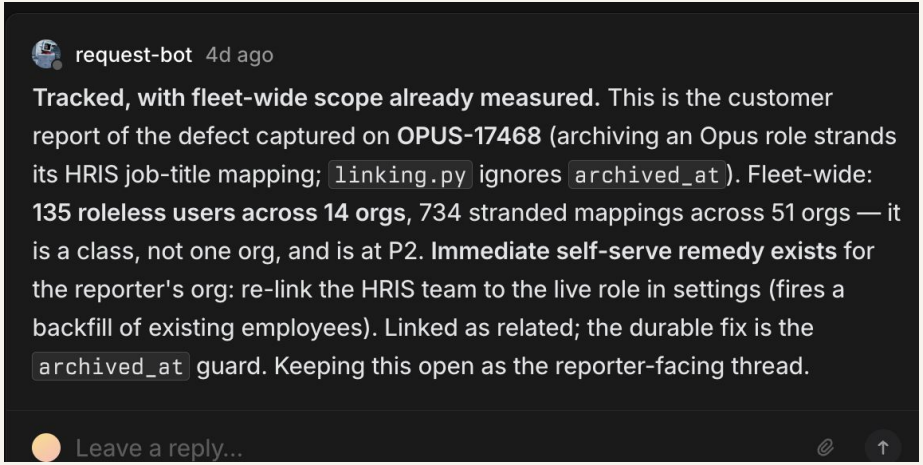
This was saving a TON of time for engineering. It *transformed* work for the CS team…

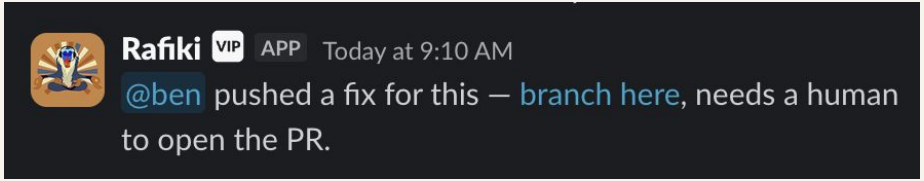
What could more could it do for us?

Triage + Open PRs

It’s able to answer questions based on code + data…
so why can’t it triage issues?

… and then fix them?





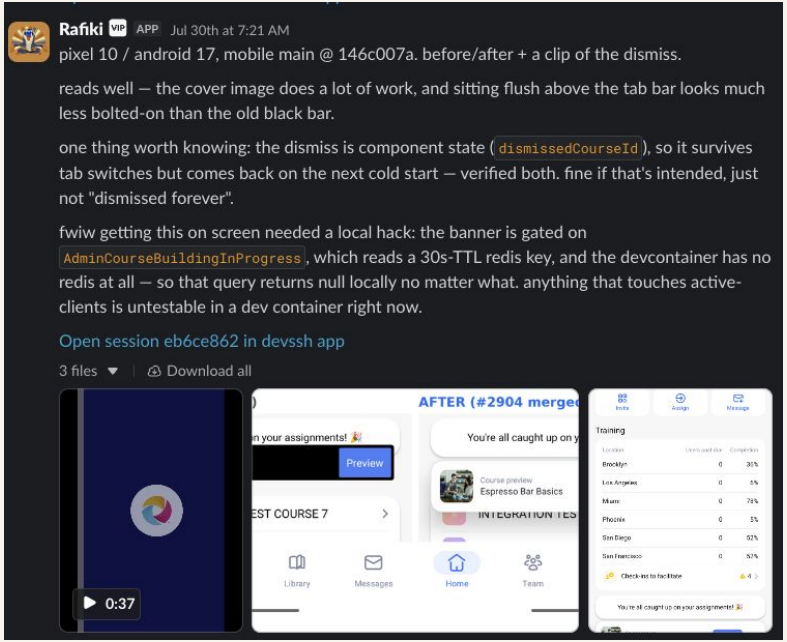
Eventually it evolved from Product Bestie into Rafiki, to match all of our other custom tooling that is named after *The Lion King*

Jan 2026

The next bottleneck was QA

PR volume grew… QA became a bottleneck

The agent spins up **the exact device** a user reported the bug on to reproduce and fix. Cross platform, video previews of every change it makes



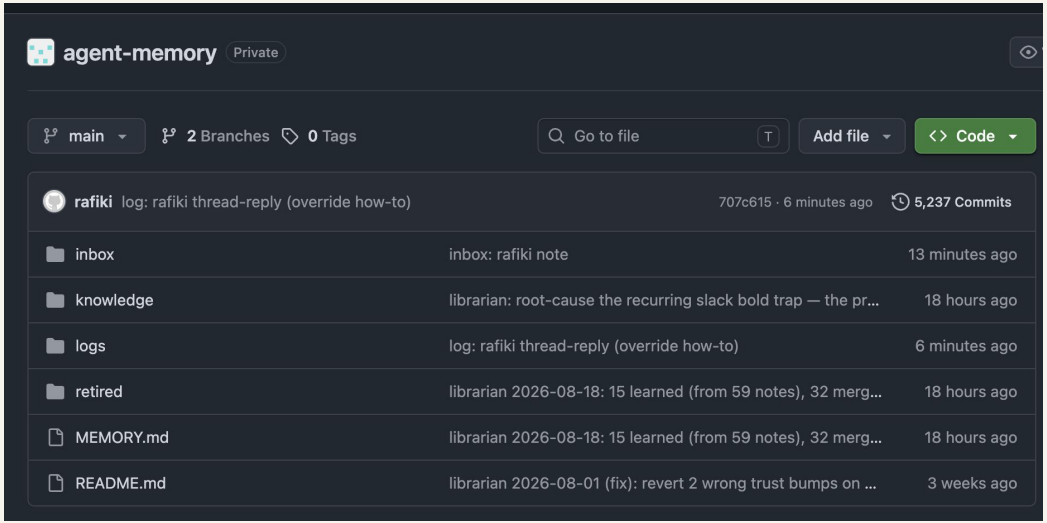
April 2026

Agents don’t have scars. They should.

An agent gets a fresh context window every session

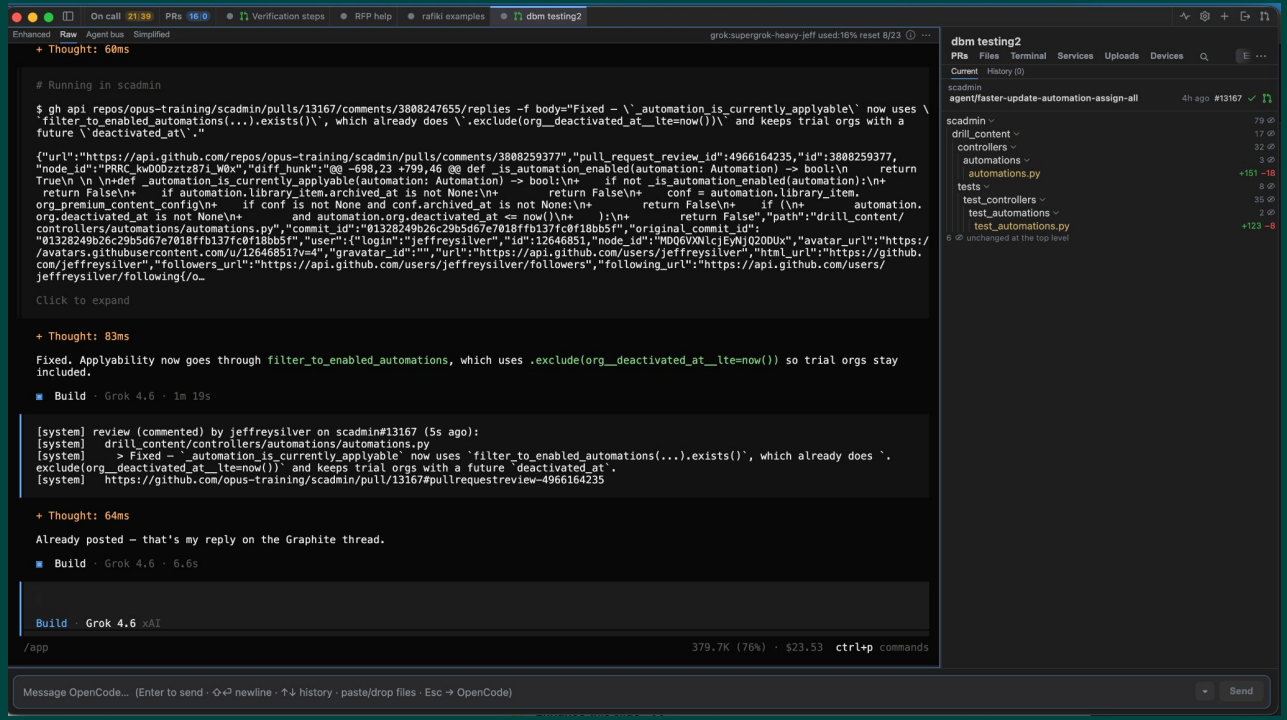
It does not feel pain or consequence. It does not “remember” what happened an hour ago

An effective on call engineer does



May 2026

Pride Rock



Built a CLI to run Claude Code in our own terminals…

had itself build it's own UI… and mobile app 😅

It's home to Rafiki

June 2026

Free time: 1% better every day, for agents too

FOUND ON ITS OWN

Rage clicks

It read the session data, found where users were fighting the UI, and shipped the fixes.

FOUND ON ITS OWN

A one-line cache change

Nobody asked for it. It was reading our own bill.

SAVED

$1,200/mo

Free time pays its own bill

July 2026

Is it working?

2,200 questions answered 2026 YTD

At 5 min per question

= ~180 saved eng hours

= $20k in savings

100 tickets closed out fully by agents

At 60 min per ticket

= ~100 saved eng hours

= $12k in savings

Bug volume down 15% YoY

despite platform usage growing by 4x

Takeaways

1. Be allergic to toil
2. Find the high leverage opportunities
3. Get to the compounding value

Thank you!

We are hiring! Find me, or email me at jeff@opus.so If you are interested!