

# The great cleanup



**This is not a technical talk. This is a feeling**

Today I'm going to share 2 stories  
I'll let you think about them yourself

The goal? There's no goal!\*

\* The goal is to (try to) make you think.

# A story where I **left**

---

# The first story is about a Startup

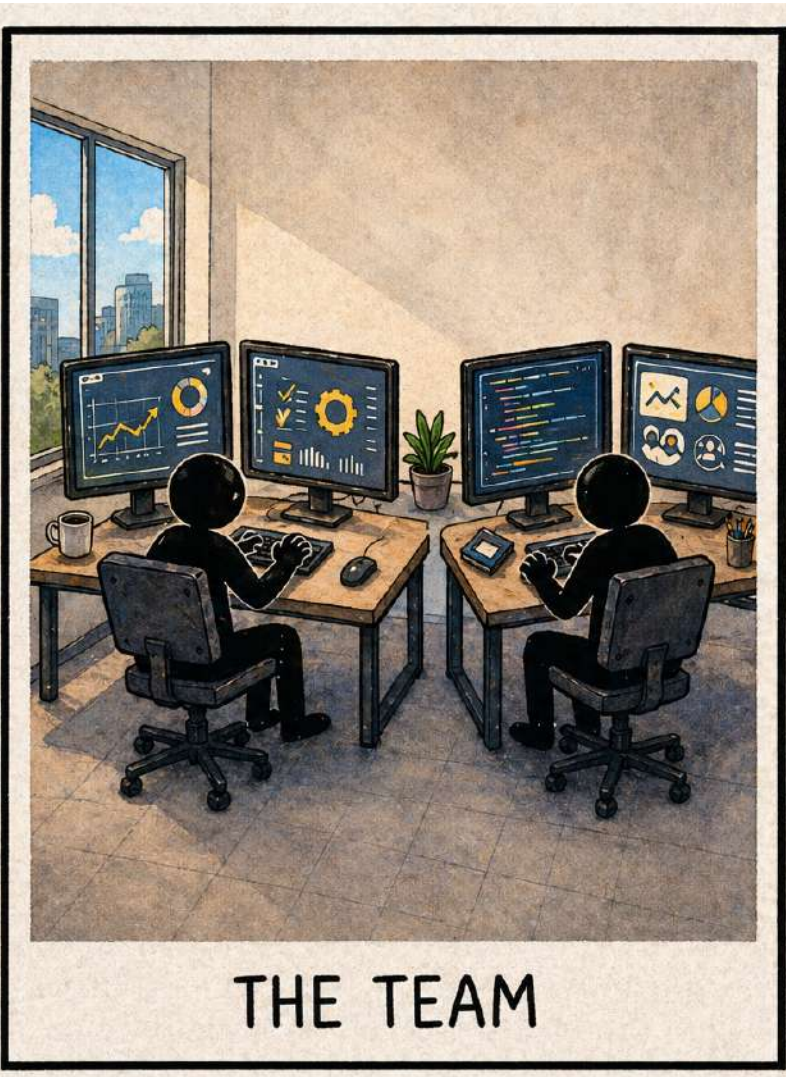
A small and passionate startup – 4 people  
Good engineers, excited founders  
Launching very soon  
I had just joined to help

**Simple web app with some automations/flows  
LLM based features for automating tedious tasks**

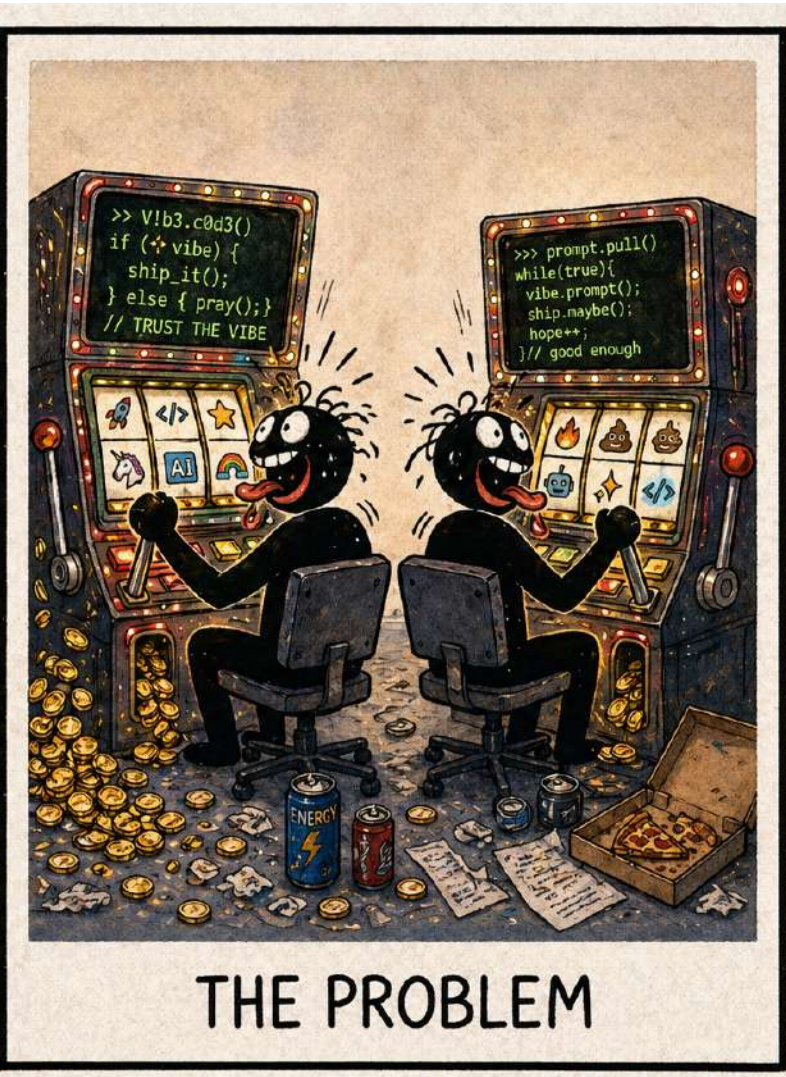
**That's it.**

**Some things were** sketchy

**Thousands of lines of useless code**  
**LangGraph pipelines for everything under the sun**  
**No stopping. Every morning a different surprise**



THE TEAM



THE PROBLEM

**And so, we launched**

**And so, we launched**

**And everything started to break**

File upload  
RAG pipelines  
Login/Out  
Parsing  
LLM  
Observability

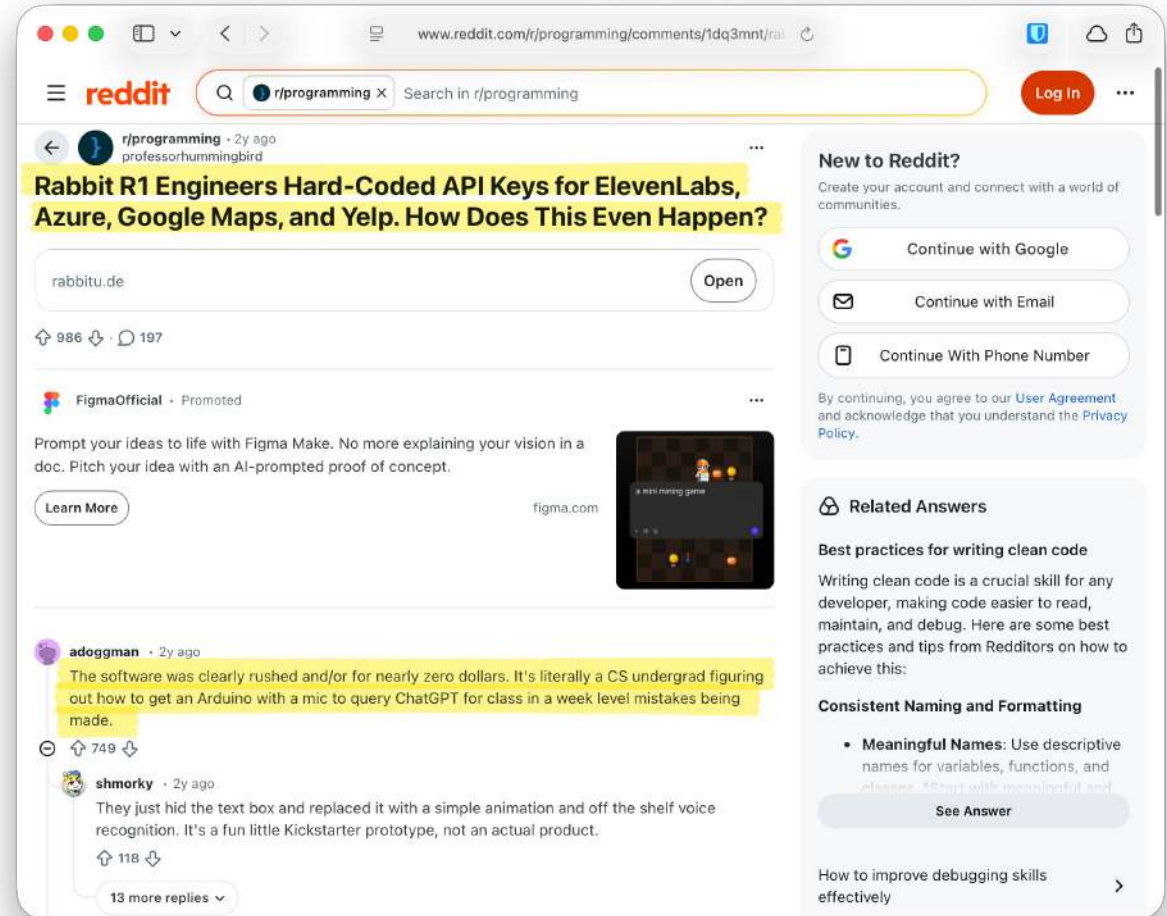
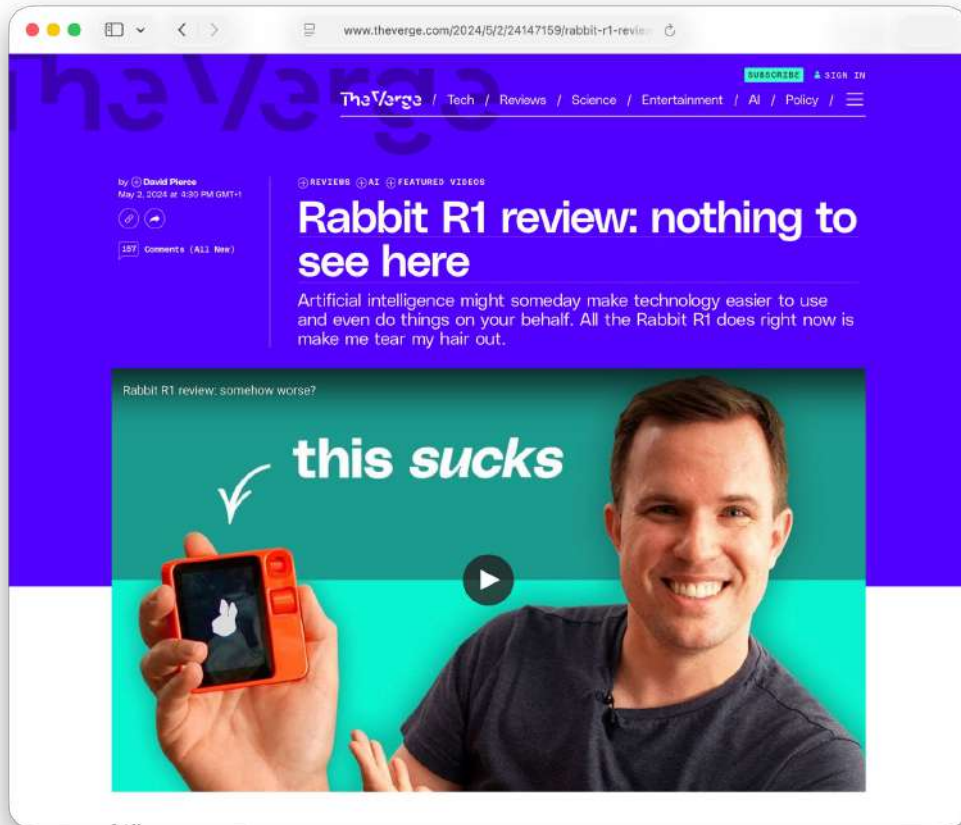


CEO



ENGINEER

**We have no time to fix things!  
We have cool things to work on!**

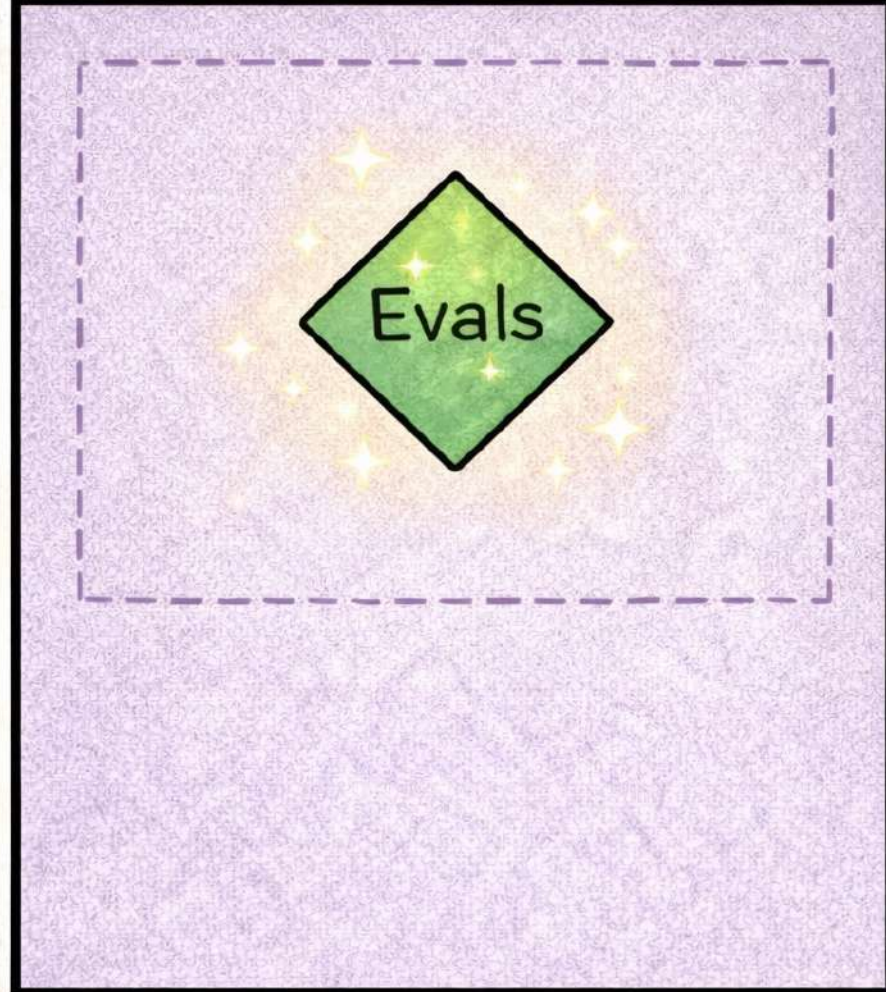


"Browser automation"

## THE SITUATION



## THE PRIORITY



What we **want** to do

What we **want** to do vs. what we **should** be doing.

**I left**

# A story where I **stayed**

---

## The second story is about another Startup

Very small team – starting to get PMF

Excited and talented founder

There's only a small issue

Our startup depends on having great data available

And we have no idea how our data pipelines work

# The relevant backstory

**WTF**

**WTF**

**WTF**

**WTF**

**WTF**

**WTF**

**WTF**

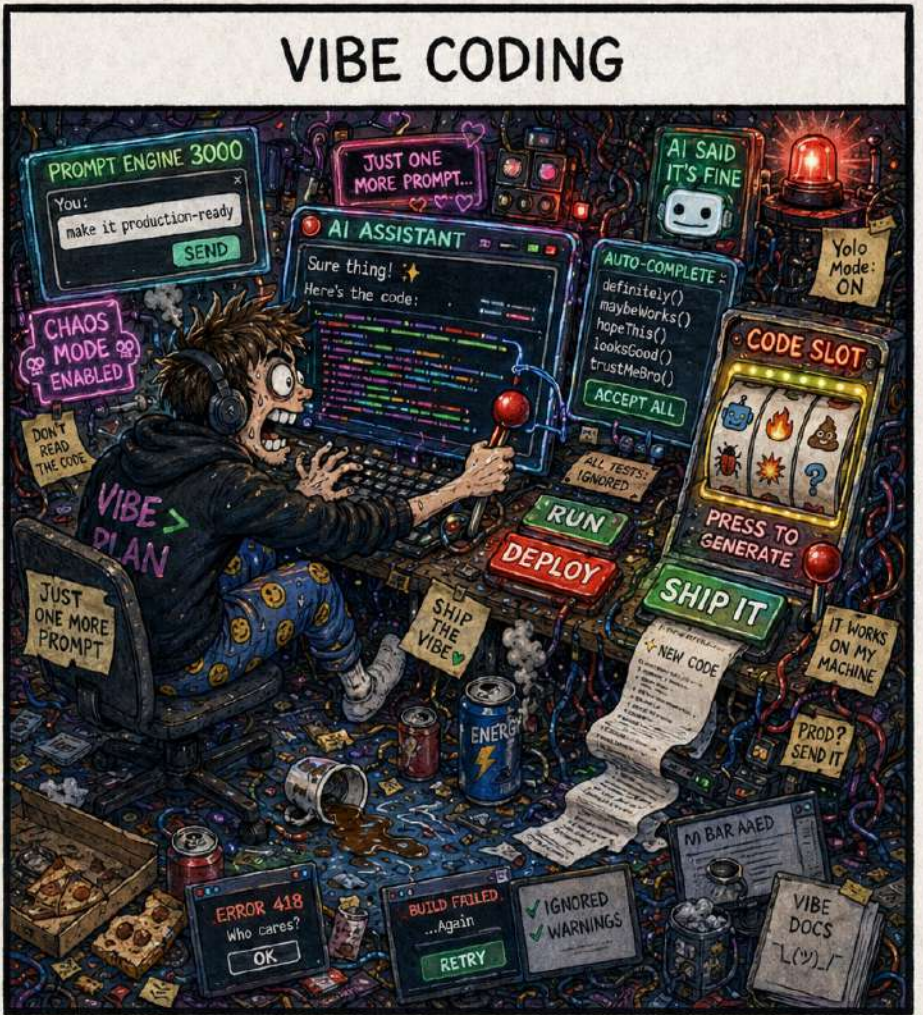
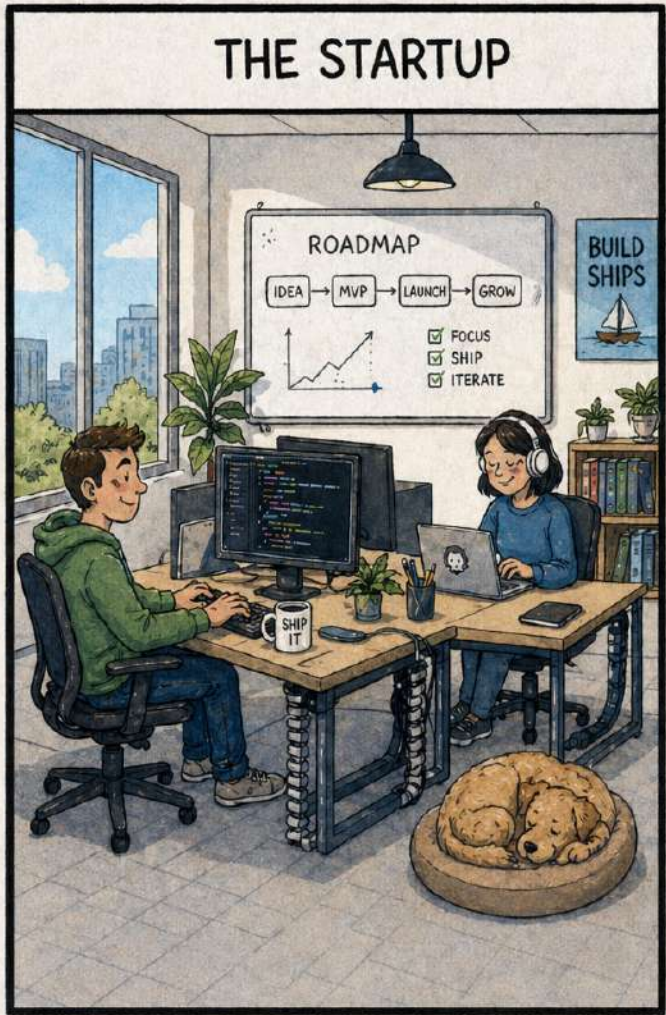
**WTF**

**WTF**

**WTF**

**WTF**

**Why are we spinning machines manually?  
Why are we using GPUs for chunking?  
How do we know things succeeded/failed?  
Why are we retrying dead links?  
Do we have an overview of how this works?  
Is the quality of the data ensured somehow?**



**And so, we spend the next 2 months fixing things.**

Infra as code, everywhere

Everything as simple as possible

Notifications on success/failure

No GPU shenanigans – just the necessary things

**Rebuilding takes more time than building**

**But the team **knew** it needed to be solved**

**I decided to stay\***

\* And fix the problem.

[duarteocarmo.com](http://duarteocarmo.com)

# Some final **reflections**

---

# A mentoring story



""What should I do now? I learned Computer Science  
What do I do? How do I learn?  
Claude writes 99% of my University projects  
The code is solved, what should I do now?  
I should really get into the AI thing  
All the code is solved""

You listened to the stories  
When was “code” the problem?

# What people think the problem is

The amount of code

The GPUs

The amount of PRs

The AI features

The langgraph and langfuse

What people think the problem is

The code

The GPUs

The amount of PRs

The AI features

The langgraph and langfuse

What the problem actually is

Building something good

Building something robust

NOT building useless things

Getting the basics right

Taste, taste is the moat

# Pulling the lever vs. ai-assisted engineering

Having good taste, thinking, and building useful things.

How will they learn?



**Nobody knows anything**

# Thank you!

