

Engineering for an LLM you don't trust.

A journey: past · present · future.

HELLO.

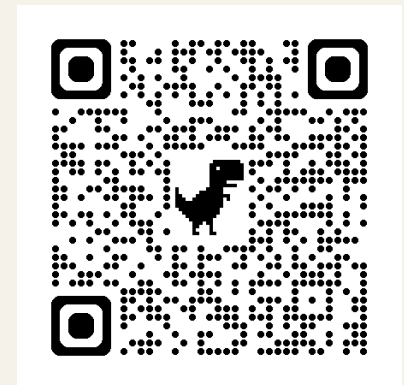
Mohamed Badawi

Software Engineer and one of the main core maintainers at Weaviate

Founder of Anonde (open-source data protection for LLM pipelines)

*Before was writing code and designing systems, now
Distrusts LLMs professionally.*

 @moogacs  @moogacs



Three acts.

I. Past.

When code was written by humans.

II. Present.

When code is written by LLMs.

III. Future.

When one engineer is a team.

PAST.

When code was written by humans.

You wrote it.

You understood it.

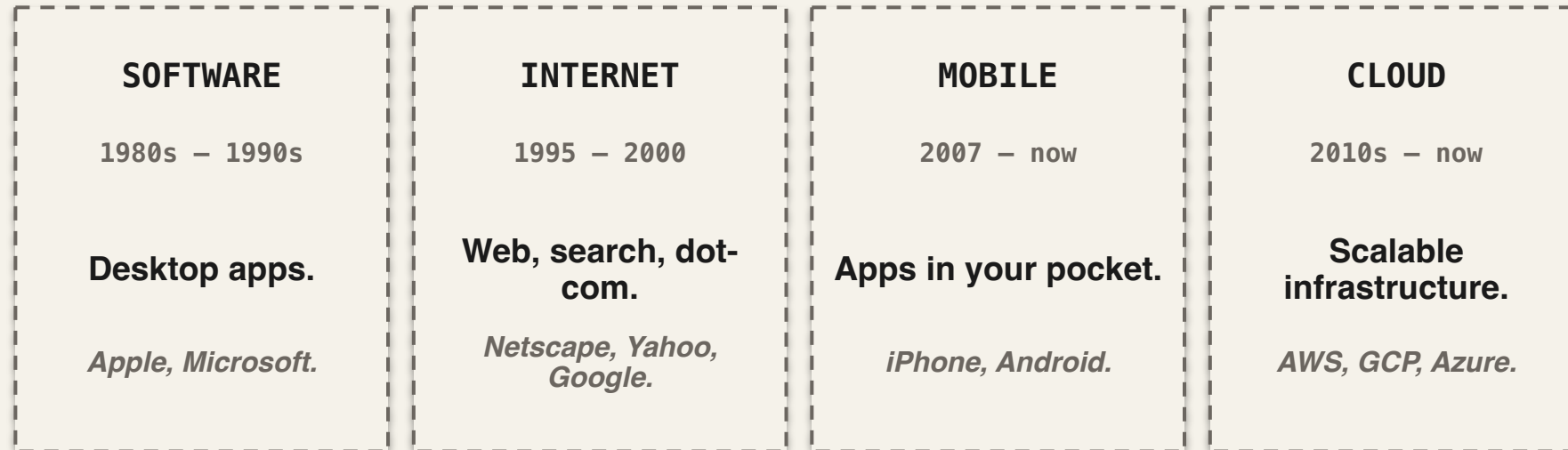
You shipped it.

**SOLID, DRY
KISS , YAGNI**

And the assumption that a human had to be able to defend every line.

Eras of software.

Each one shorter than the last but lasted in development



Adoption cycles got shorter.

PRESENT.

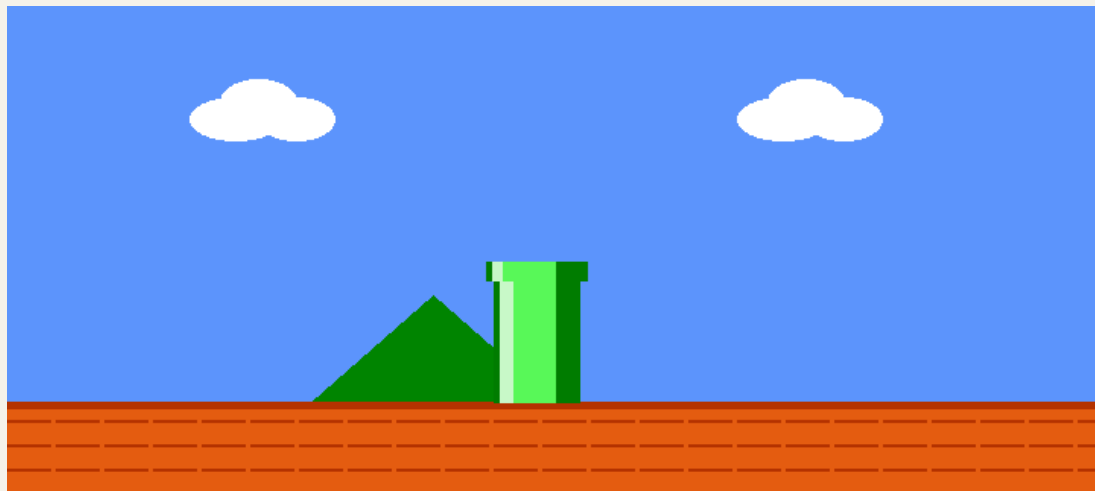
When code is written by LLMs.

Same engineer. Same obstacles.

What changed wasn't the work, it was the pace.

2020

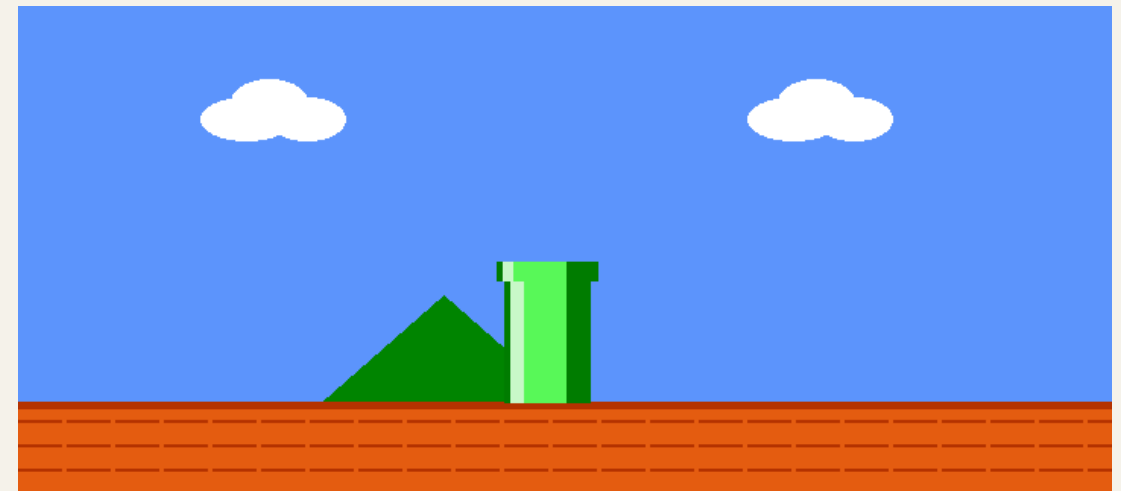
One feature, one week



Normal pace.

2026

Same feature, an hour.



4x faster. Same engineer.

Panic every where

deathbyclawd.com

ARE YOU JUST A .md FILE?

The SaaSocalypse Survival Scanner

Find out if your SaaS can be replaced by a Claude Skill

hubspot.com

SCAN FOR DEATH

\$285B+ market cap wiped · ∞ .md files generated · 0 survivors

RECENT: amazon.com 4 asana.com 78 atlassian.com 52 bauwerk-group.com 8 lemmy.ml 1

📄 QQQ Top 50 (50)

☁️ Cloud Index (64)

🇪🇺 Euro Tech (30)

By Death Score

A-Z

Verifying Connec... \$TRST
trustpilot.com

72

FLATLINING

"A billion-dollar business built on strangers arguing in the..."

Digital Transformati... \$SOW
softwareag.com

52

CRITICAL

"Software AG spent 55 years building enterprise middlewar..."

Attention Required! \$SGE
sage.com

42

SWEATING

"Sage has been selling the same accounting software since 198..."

TeamViewer \$TMV
teamviewer.com

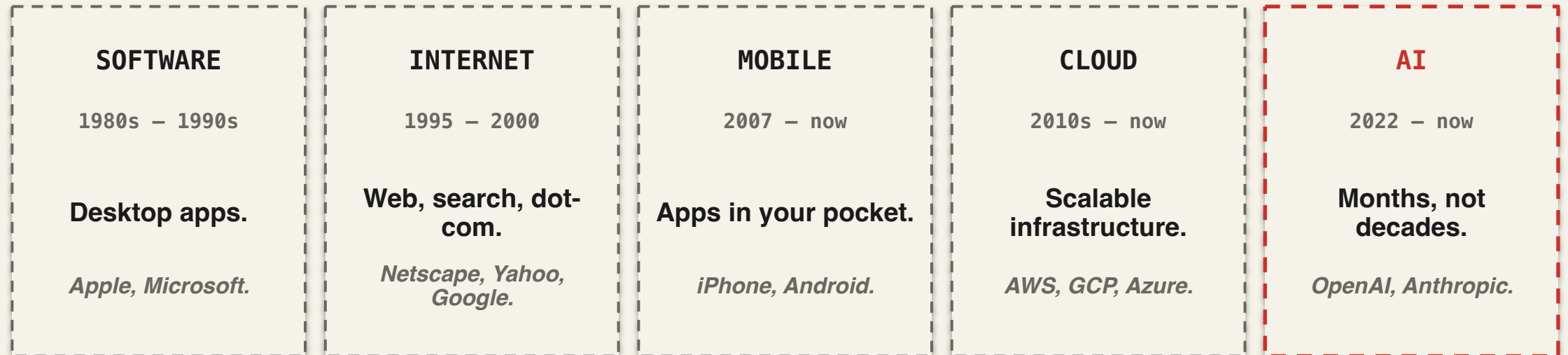
31

SWEATING

"TeamViewer charges enterprises \$50/month to mov..."

Five eras of software.

Each one shorter than the last.



Adoption cycles got shorter. AI compressed from decades to months.

We're standing in the shortest, fastest era yet.

PRs and diff got bigger.

Reviews became longer.

Ownership is degrading.

Understanding got demoted.

Chaos is the mainstream

What happened ?

Summer of Bliss

The curl project will not accept or otherwise handle any vulnerability reports during the month of July 2026. We call it the *curl summer of bliss*.

curl's [submission form on Hackerone](#) will be paused starting July 1, 2026.



Summer of bliss starts: **July 1, 2026. 00:00 CEST**

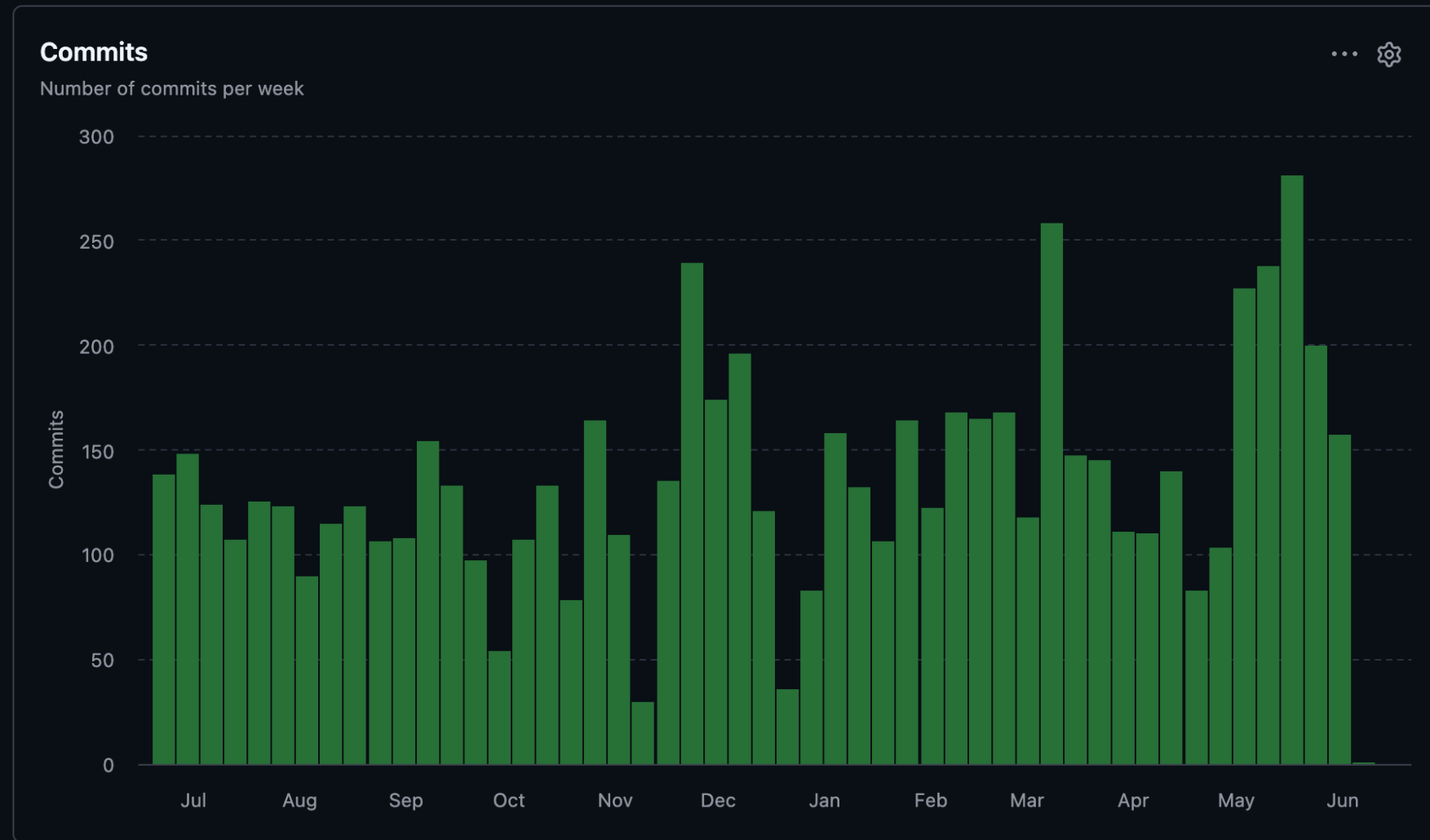
Submissions resume: **August 3 2026. 09:00 CEST**

<https://daniel.haxx.se/blog/2026/06/15/curl-summer-of-bliss/>

Weaviate

I knew every callback. I could explain every lock. I almost remember every incident

Commits over the last year of weaviate/weaviate



May 14, 2026 – June 14, 2026

Period: 1 month

Overview

324 Active pull requests

65 Active issues

256
Merged pull requests

68
Open pull requests

28
Closed issues

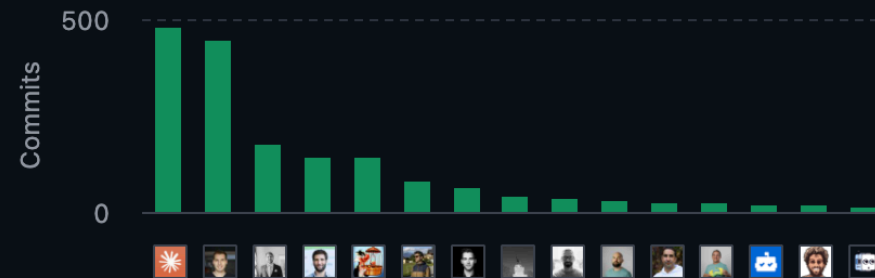
37
New issues

Summary

Excluding merges, **34 authors** have pushed **526 commits** to main and **1271 commits** to all branches.

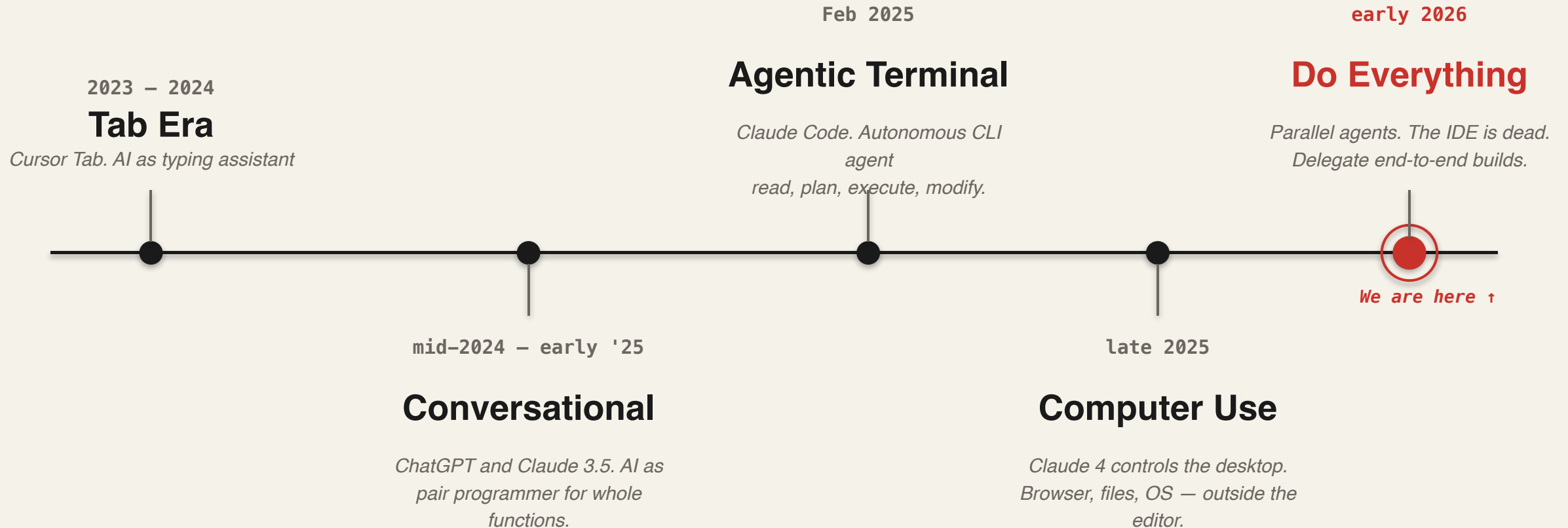
On main, **1808 files** have changed and there have been **315,124 additions** and **32,476 deletions**

Top Committers

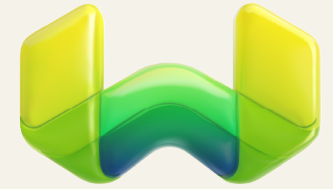


How AI rewrote the IDE.

Three years. Five eras. We're still going.



From tab completion to autonomous agents – in 36 months.



**The LLM in your pipeline
is a brilliant intern
with no memory
and too much confidence.**

The one in your IDE is the same intern. Faster.

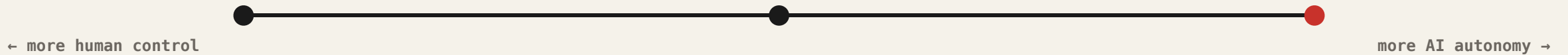
**I am no longer
the author of my code.**



I am the editor.

Three kinds of engineer.

How developers relate to AI from strict human control to full delegation.



The Skeptic

RISK AVERSE

PHILOSOPHY

Prefers strict human control. Focuses on system entropy and the degradation of institutional knowledge.

PAIN POINTS

- High fear of data-security leaks.
- “Black box” logic makes legal and technical compliance difficult.
- AI-generated bugs look perfectly correct but are highly unreliable.

The Evaluator

CAUTIOUS ADOPTER

PHILOSOPHY

“Trust, but verify.” Uses AI as a leverage tool but keeps heavy human oversight and testing.

PAIN POINTS

- Time saved on pure coding is often offset by longer review cycles.
- Fragmented data and poor documentation limit AI’s precision.

The Enthusiast

AI ORCHESTRATOR

PHILOSOPHY

Believes in “vibe coding” and agentic workflows to maximize output velocity and architectural scope.

PAIN POINTS

- Risk of shipping code faster than engineers can read or understand.
- Loss of hands-on coding skill as the job shifts to AI orchestration.

Most engineers are a blend and the whole field is sliding right.

Remember Pair programming ?

the *driver*, writes **code** while the other, the *observer* or *navigator*, **reviews** and considering the broader architectural picture.

FUTURE.

When one engineer is a startup.

Software engineering.

“Software engineering' encompasses not just the act of writing code, but all of the tools and processes an organization uses to build and maintain that code over time. Software engineering can be thought of as 'programming integrated over time”

– Software Engineering at Google. https://en.wikipedia.org/wiki/Software_engineering#Definition

Or, honestly:

Problem solving.

Everything else is implementation detail.

REMEMBER THE DEFINITION.

The job was never the code.

It's the problem.

AI doesn't replace us.

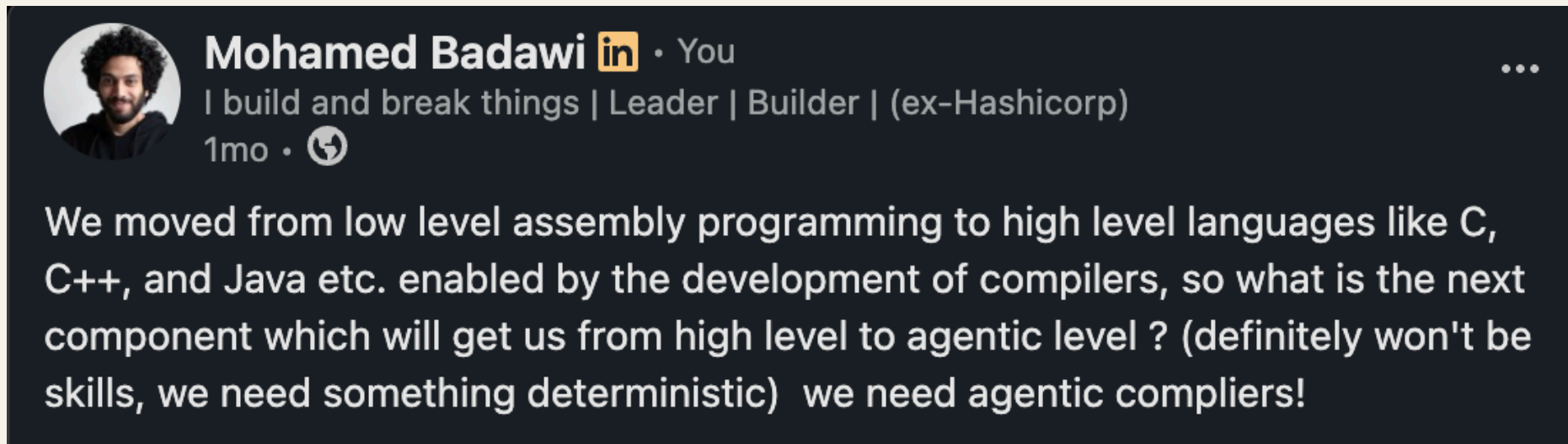
It hands us the leverage.

Don't be skeptical. Be early.


The engineer's job shifts from author to judge.

Reading > Writing. Verifying > Producing.


Is this is the end ?



A screenshot of a LinkedIn post by Mohamed Badawi. The post features a circular profile picture of a man with curly hair and a beard. The text of the post discusses the evolution of programming languages from assembly to high-level languages like C, C++, and Java, and asks for the next step towards an 'agentic level'.

Mohamed Badawi  · You ...

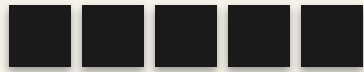
I build and break things | Leader | Builder | (ex-Hashicorp)

1mo · 

We moved from low level assembly programming to high level languages like C, C++, and Java etc. enabled by the development of compilers, so what is the next component which will get us from high level to agentic level ? (definitely won't be skills, we need something deterministic) we need agentic compliers!

What a team shipped in 2020, one engineer ships in 2026.

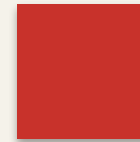
2020



team of N



2026



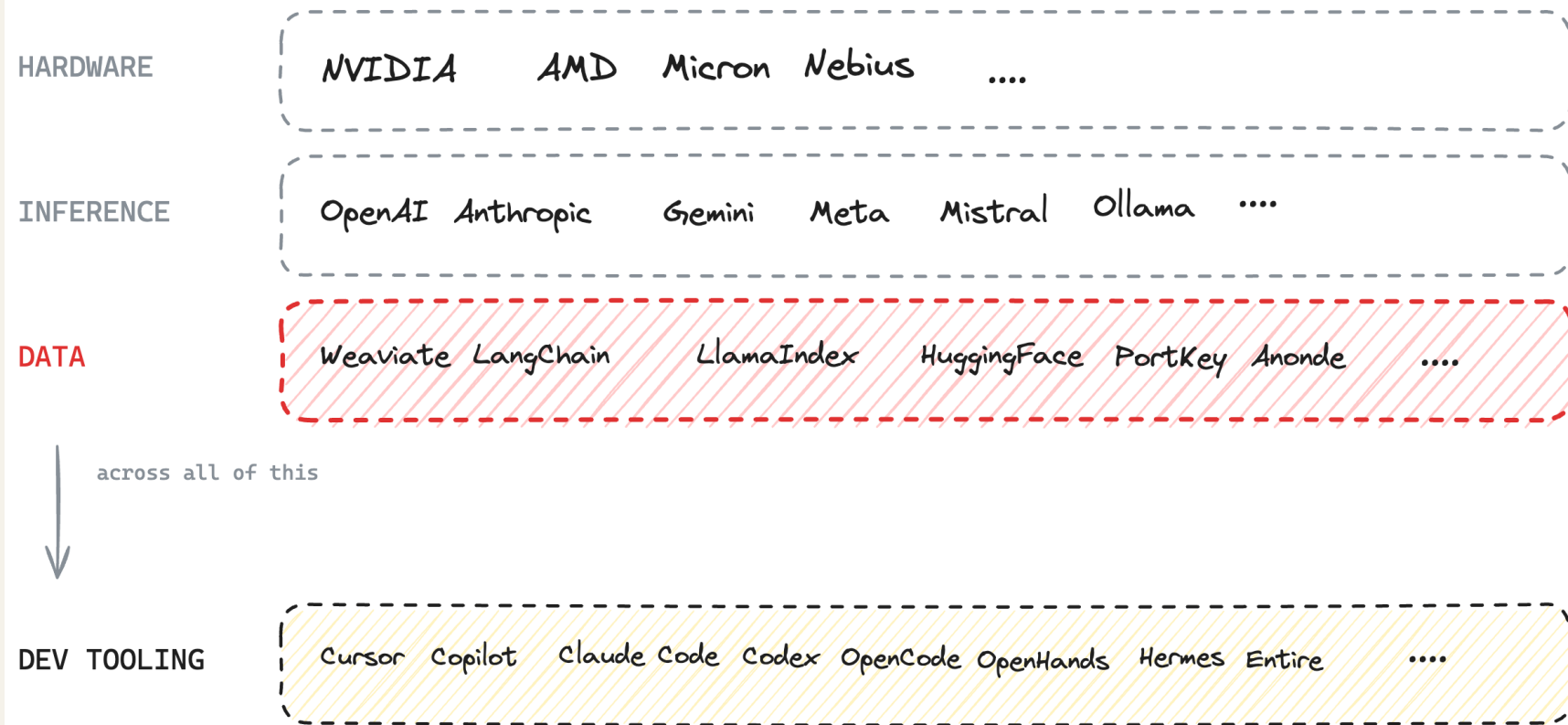
one engineer

Same product. Same scope.



AI is three layers.

Two are consolidating. One is wide open. One wraps them all.



Three layers. Plus the toolkit that touches them all.

The boring layer.

Deterministic guardrails around non-deterministic systems.

Vector stores. Schema validators. Output checkers. Tokenizer guards.

PII redactors. Tool sandboxes. Cost limits.

The unsexy work somebody still has to do.

Anonde came from paranoia.

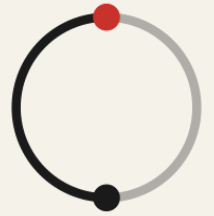
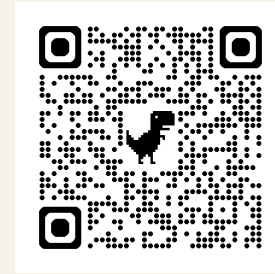
Compute commoditizes.

Reasoning commoditizes.

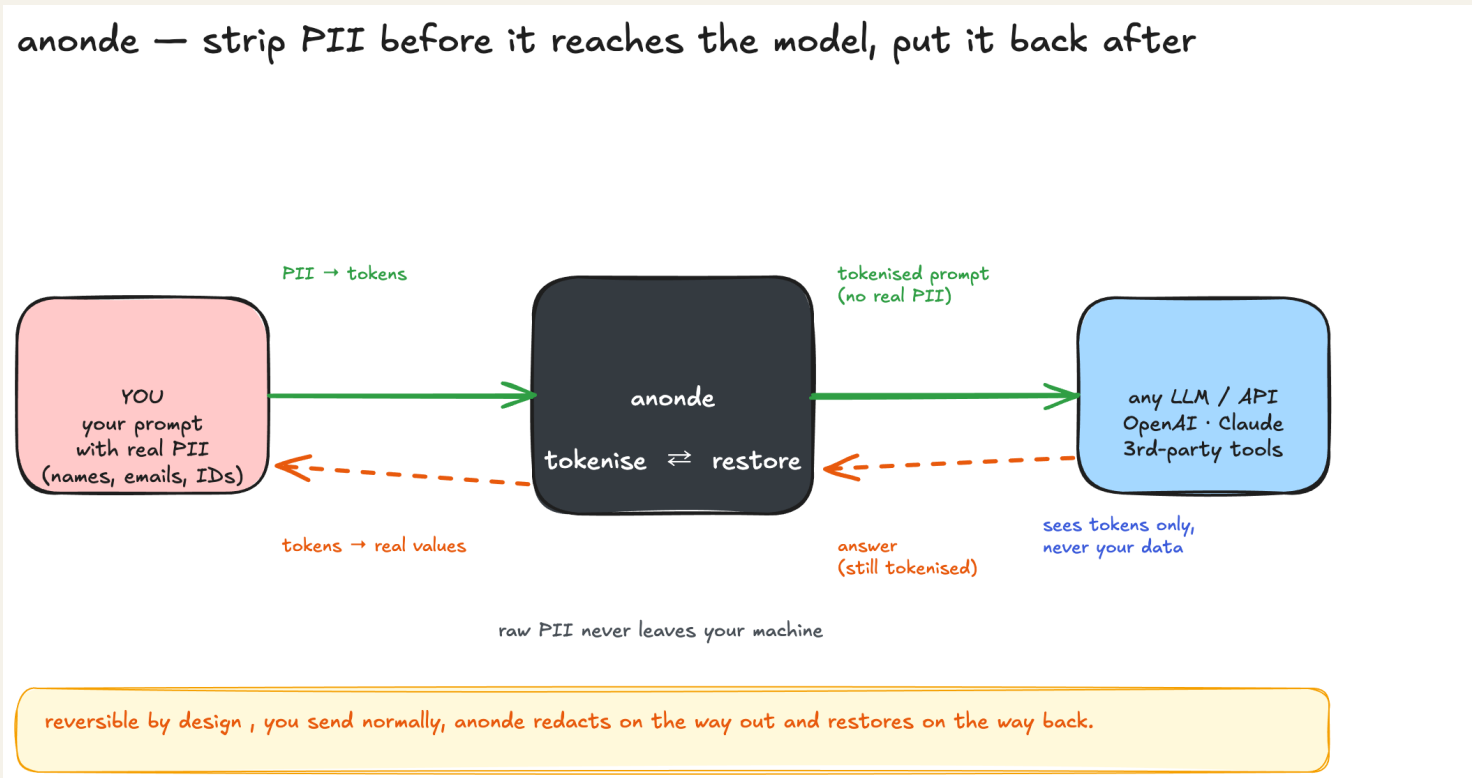
Data doesn't.

<https://anonde.io/>
github.com/anonde-io/anonde

Anonde.



Open-source data protection for LLM pipelines locally.



Demo

Harness ?

“TLDR: Agent = Model + Harness. Harness engineering is how we build systems around models to turn them into work engines. The model contains the intelligence and the harness makes that intelligence useful. We define what a harness is and derive the core components today's and tomorrow's agents need.”

<https://www.langchain.com/blog/the-anatomy-of-an-agent-harness>

Or, Simply: Harness means the surrounding parts that enables the model to accomplish tasks in the real world.

Workflow, Framework, A System



**Vibe
coding**



**Agentic
Engineering**

**Every engineer
is enabled
like never before.**

Barrier to entry became so low

Rules.

Use AI without losing yourself.

01 Use AI as an assistant.
Not as your leader.

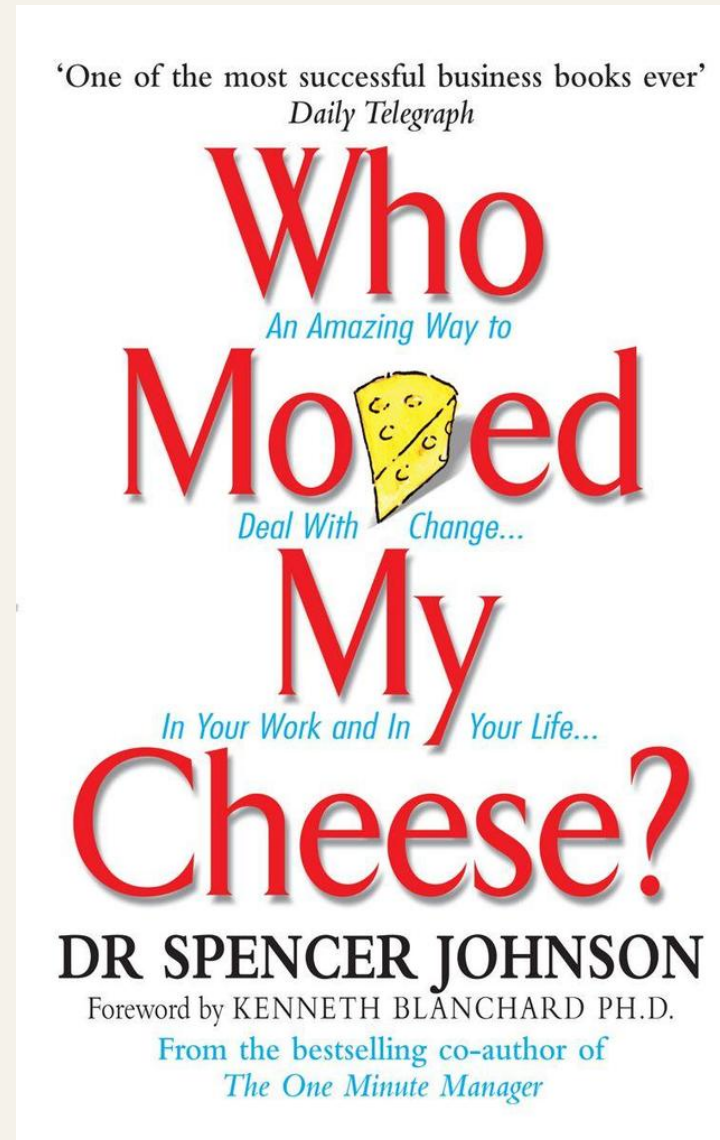
02 Use AI to understand more.
Not to think less.

You're still in the driver's seat.
The model writes. You decide.

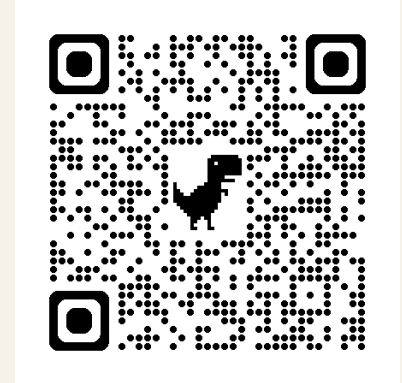
03 Use AI to ship faster.
Not to skip the thinking.

04 Don't forget it's trained on average
Aim for best not average.

05 Ask why ? What if ? Be critical
Critical thinking is the most valuable skill right now.



**That's it.
Go build.
Thank you.**



FIN.

Q&A

