

TACKLING THE JUNIOR DEVELOPER CRISIS IN THE AGE OF AI

How to learn when AI makes
everything too easy

Giordano Scalzo

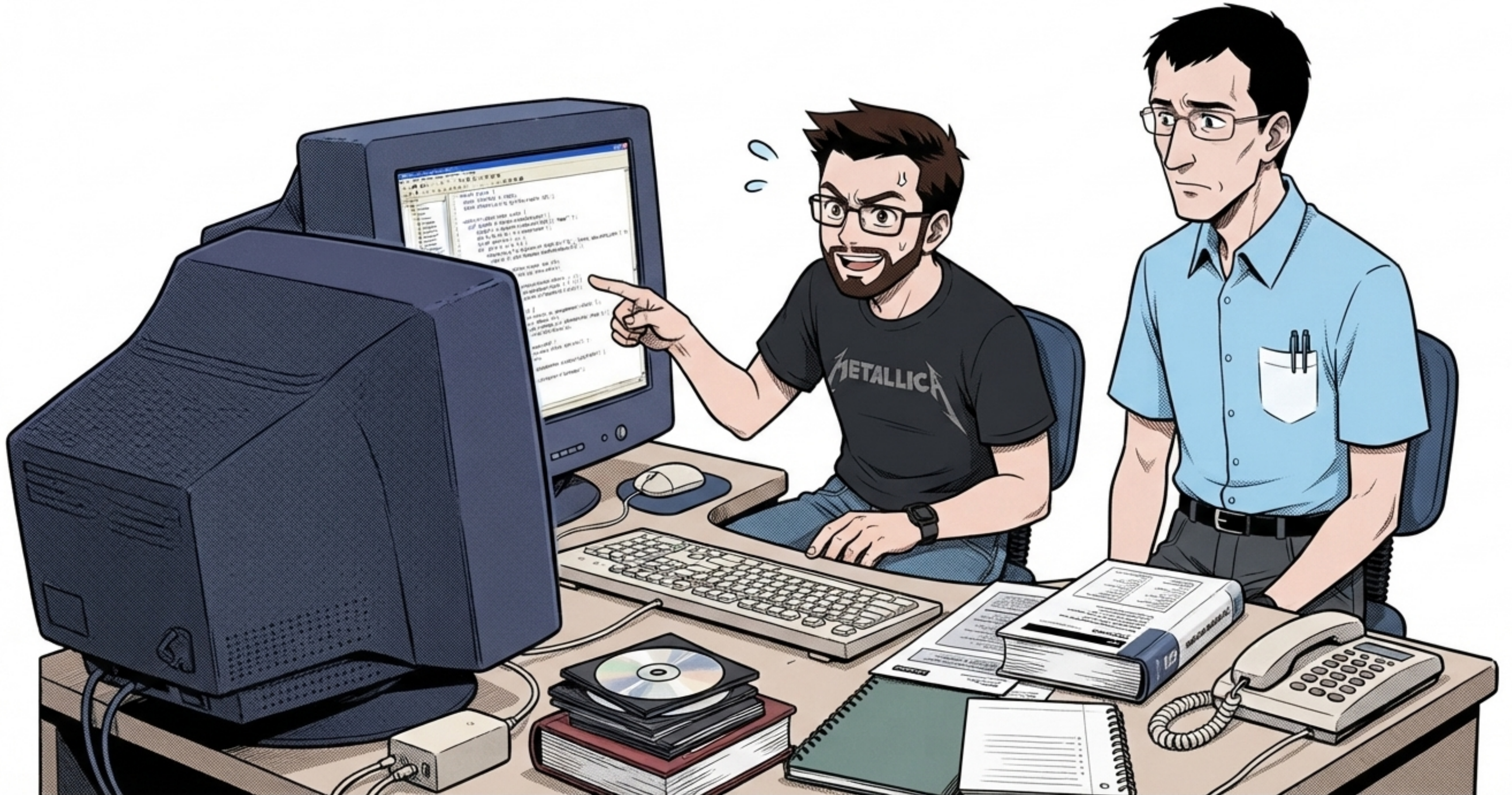


ACT ONE

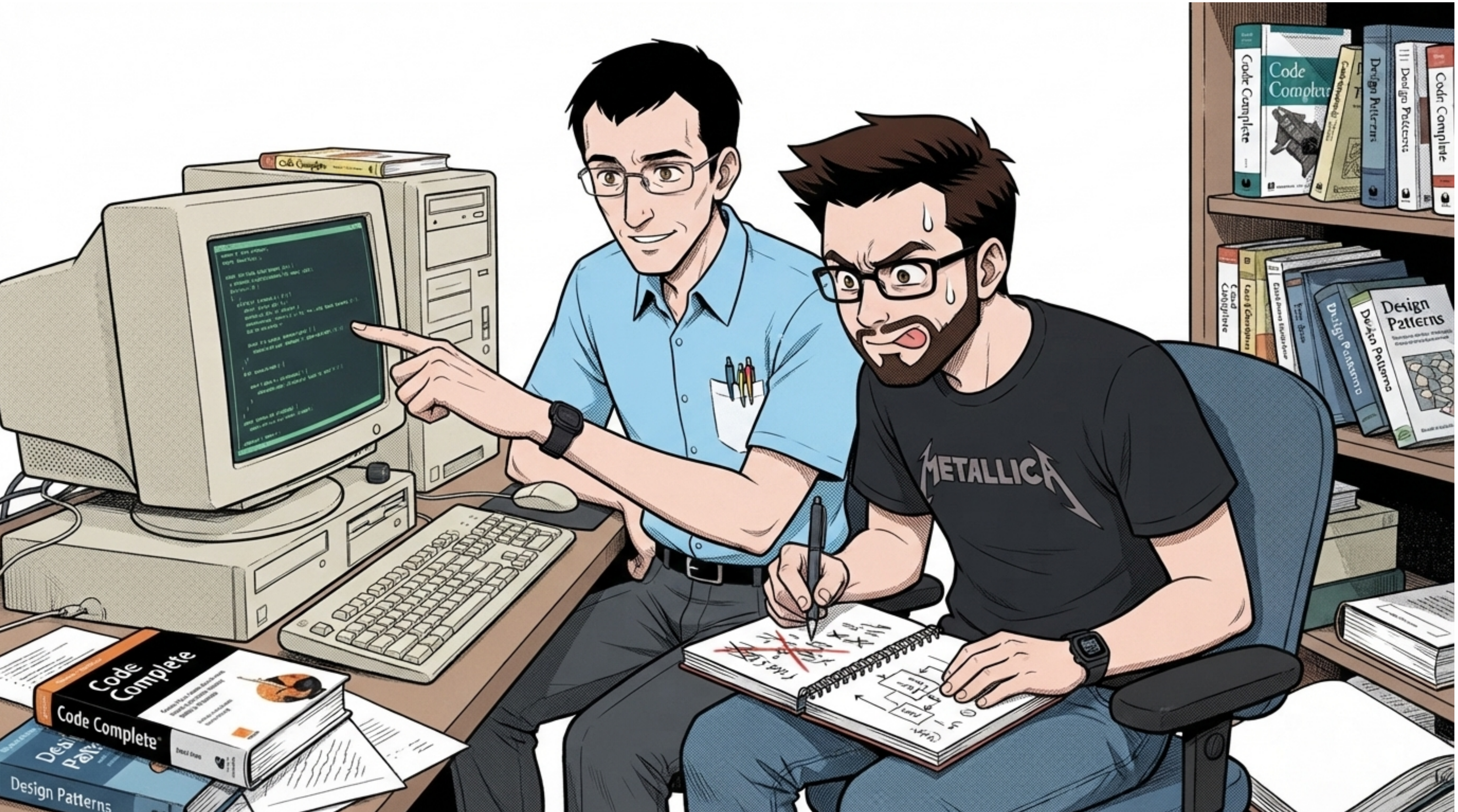
ONCE UPON A TIME



Max, my mentor



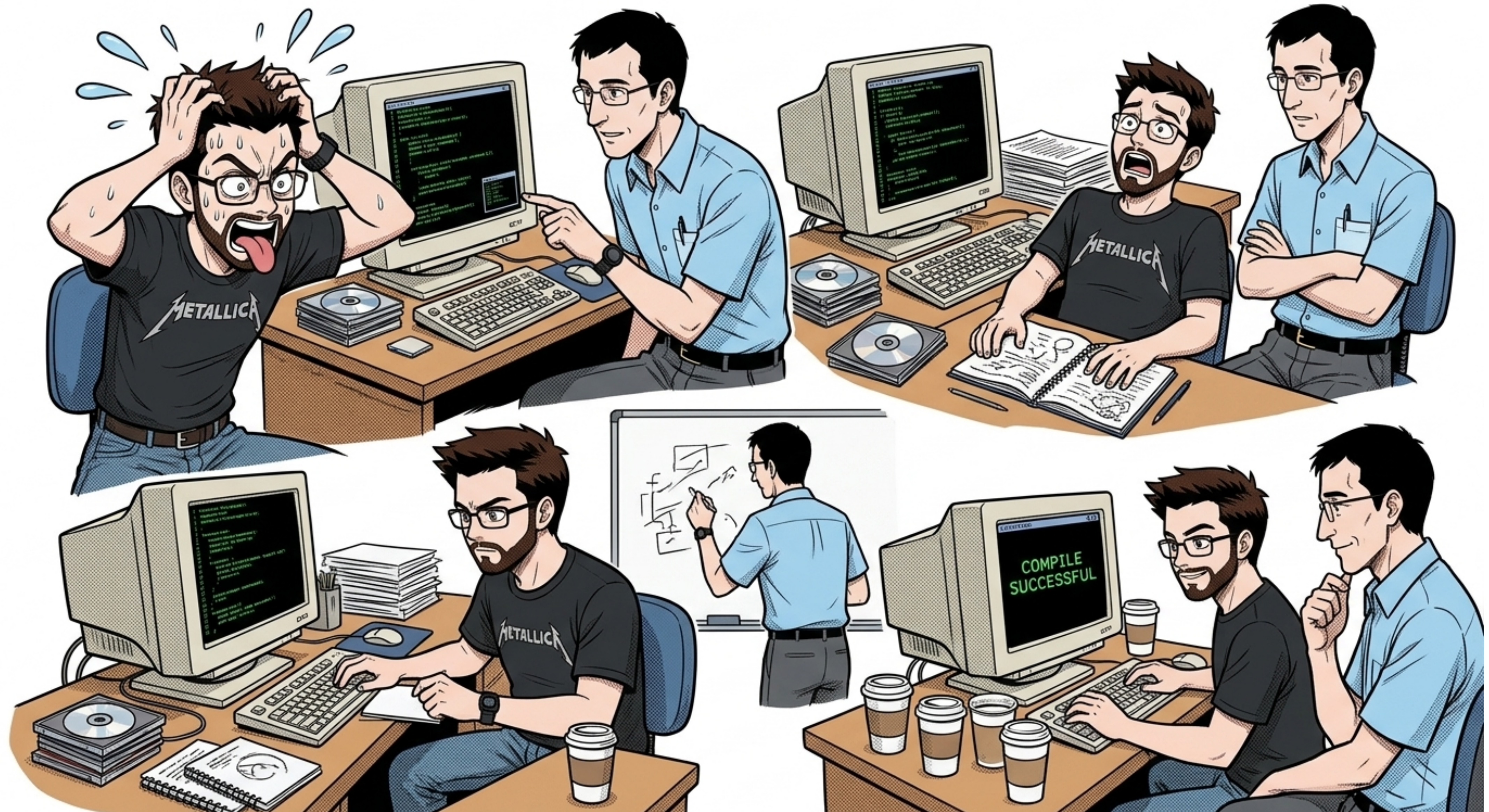
What Good Looks Like



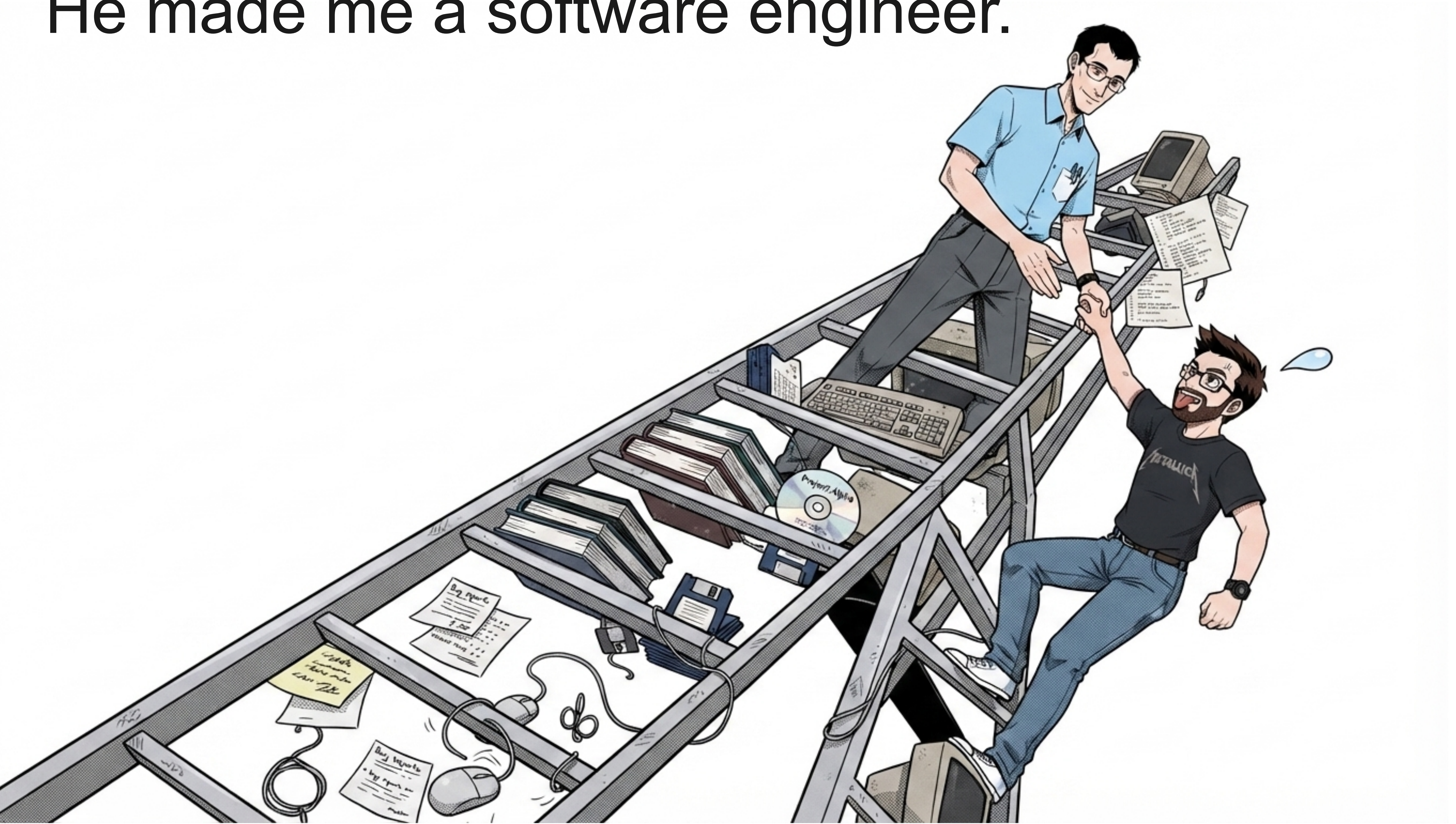
The boring tasks that nobody wanted to do



The mentoring loop



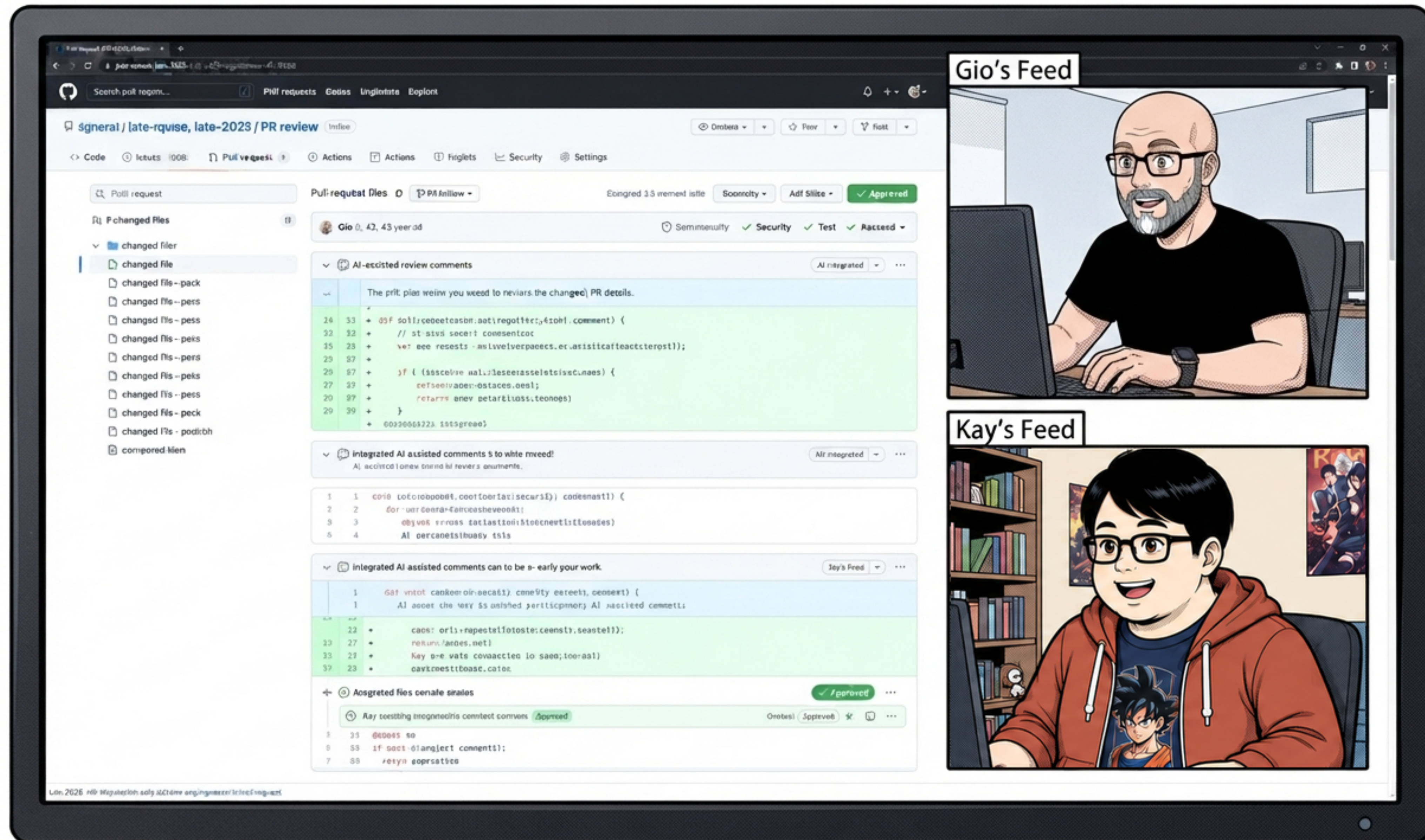
He made me a software engineer.



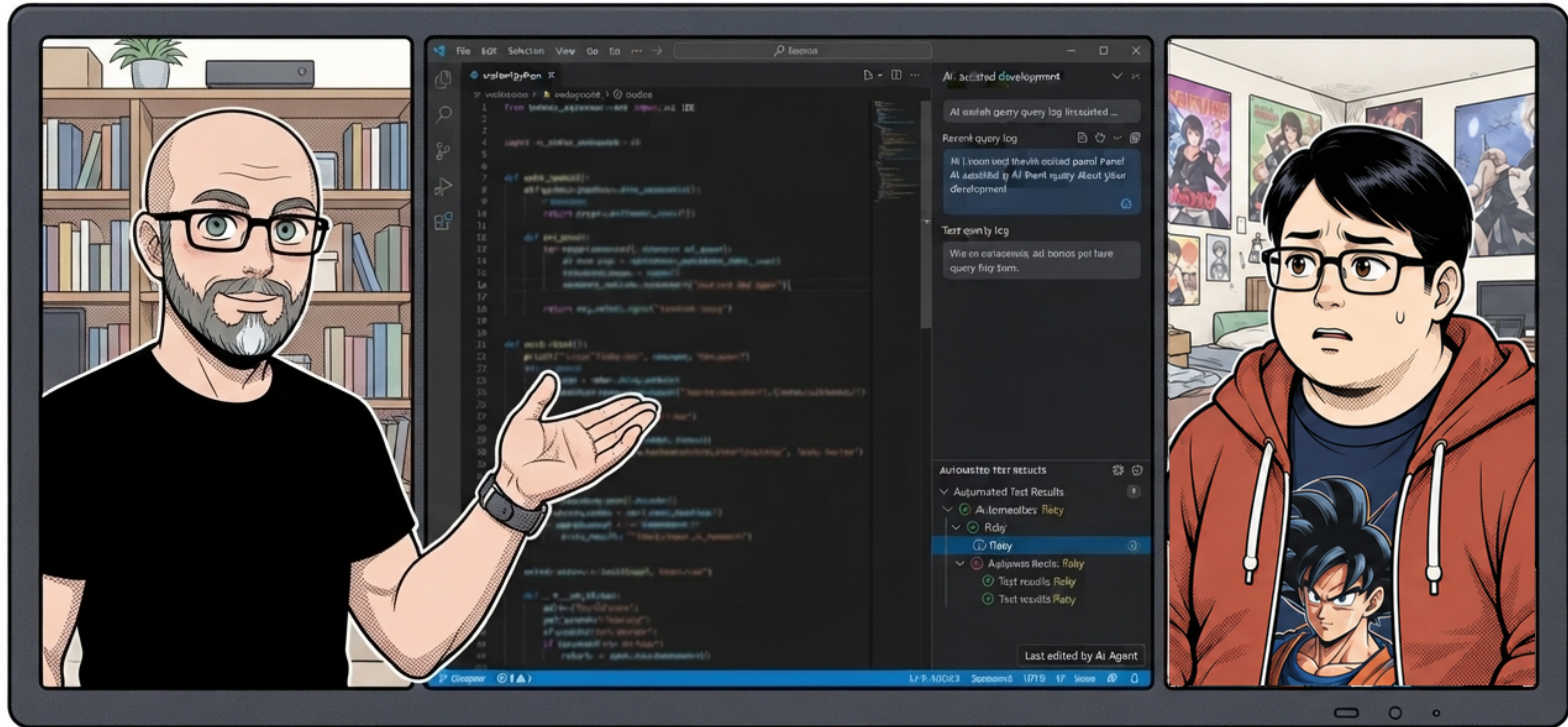
A few decades later....



A complicated feature in a few hours



How does it work?



The quiet: the confidence gap



The quiet: the confidence gap



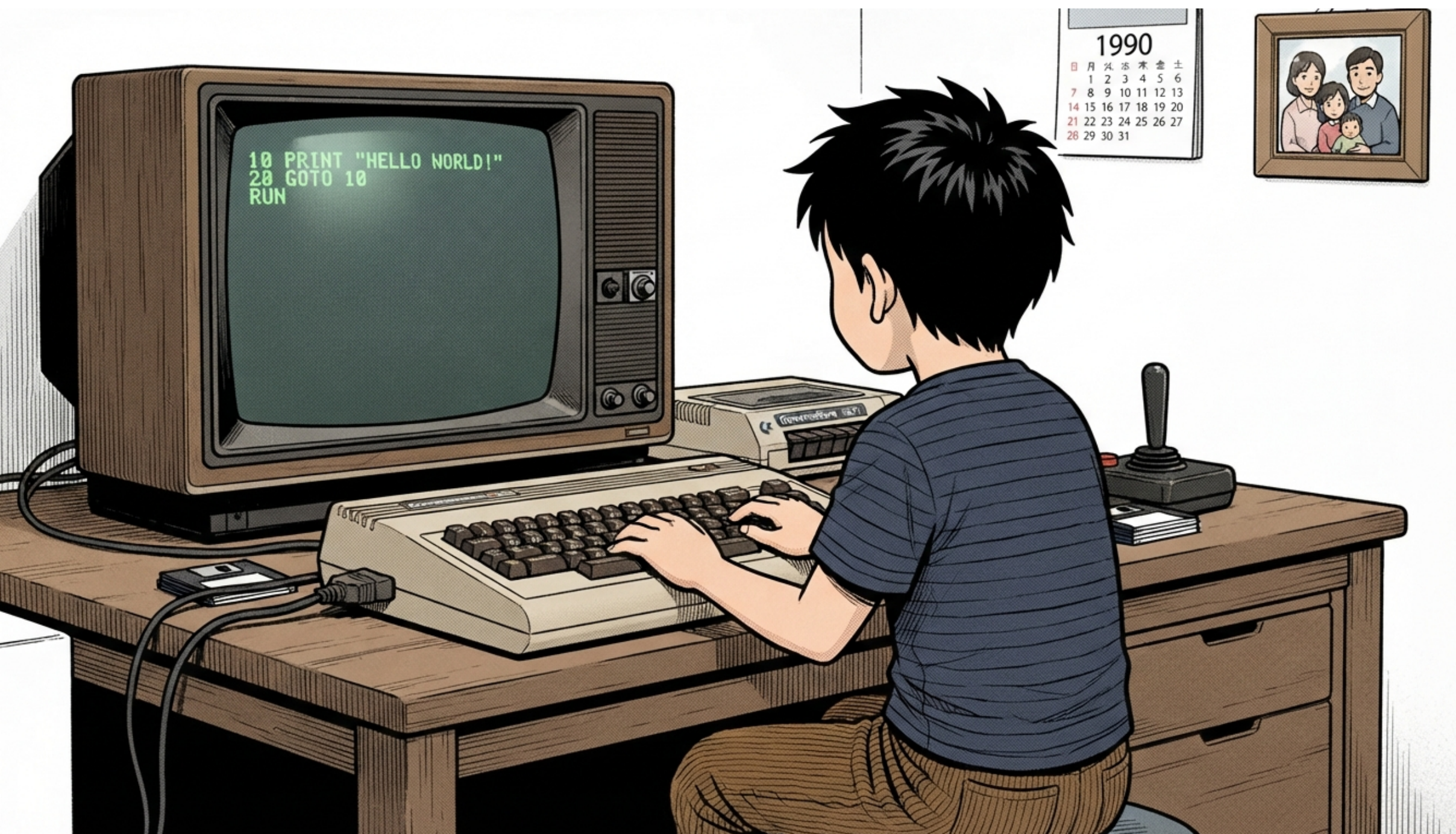
I HAVE A PROBLEM



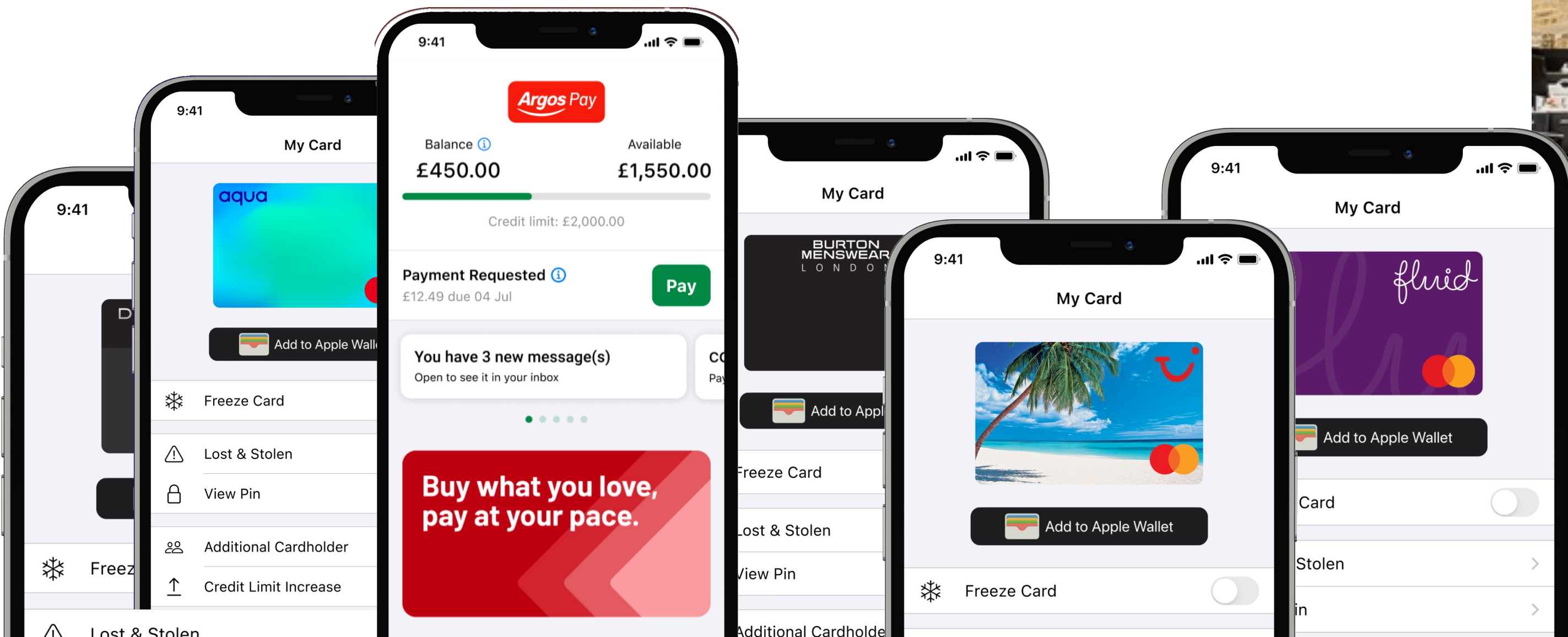
† **WE** HAVE A PROBLEM



I'm Gio, and I work with
computers.



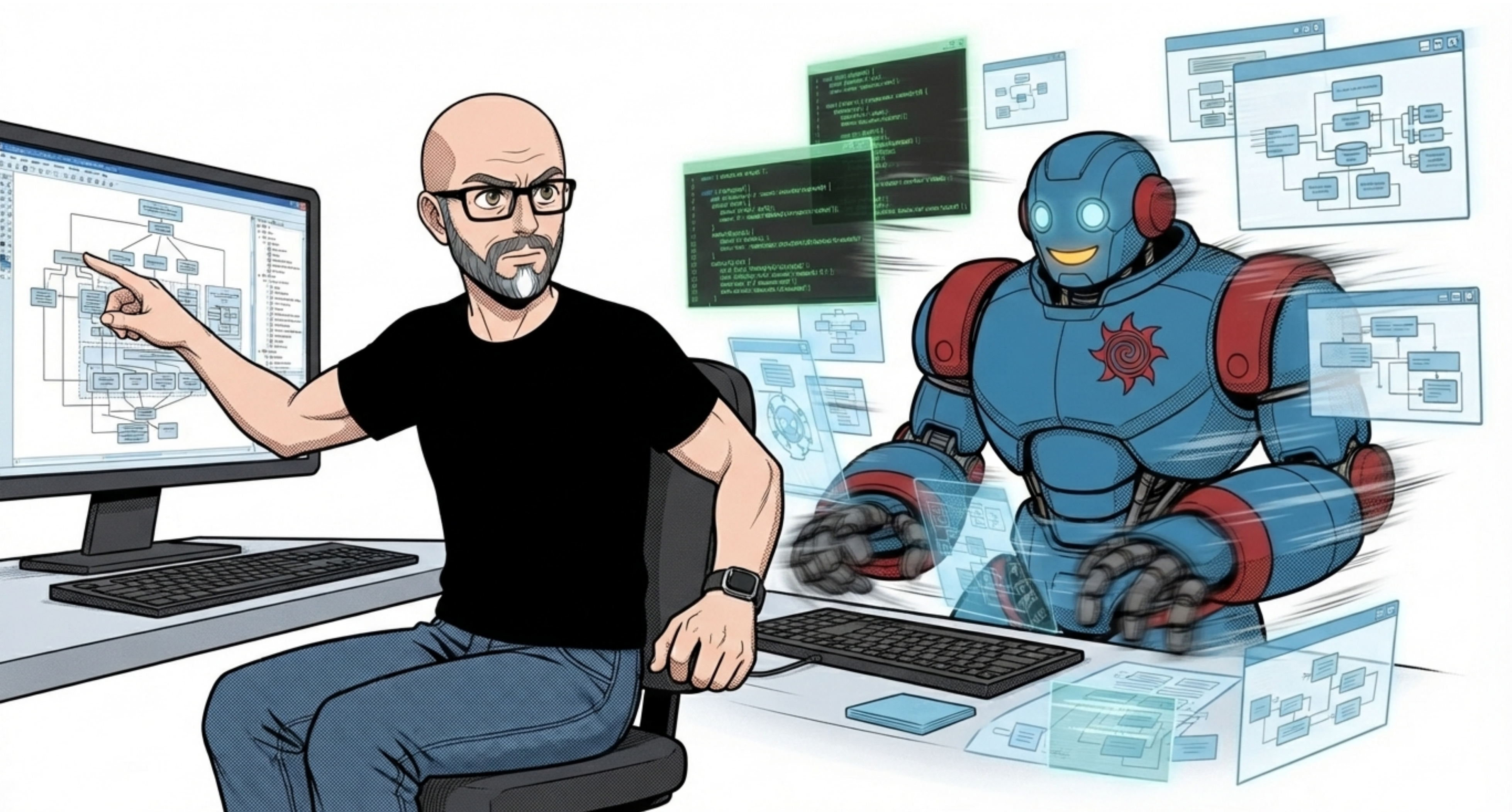
London, NewDay, Mobile



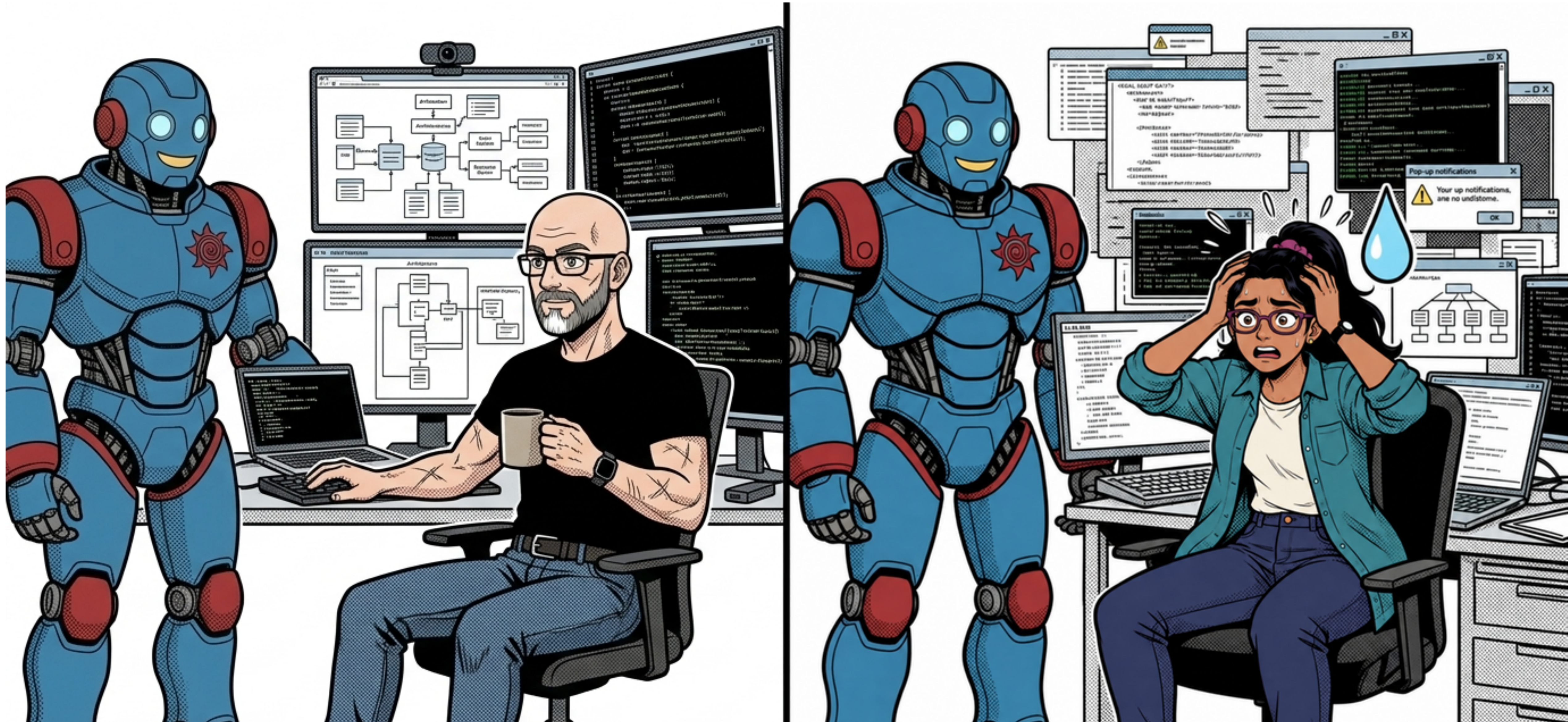
A varied career across roles and companies



I'm an AI enthusiast.



A junior cannot. Same output, different judgment.



The LeadDev

Engineering Leadership Report 2026

How engineering leadership is evolving,
and what it means for the future of the role.



Key findings

84% believe AI will make it harder for junior developers to enter and grow in the profession.

41% say their team is less motivated than 12 months ago.

45% of leaders are working more hours than this time last year.

37% of engineering leaders are doing more technical work.

AI for internal use is now the #1 engineering priority – up from

45% to 77%

One in three managers are considering **ditching management** to become an individual contributor.

The traditional on-ramp to engineering – writing code, making mistakes, learning from review – is being systematically altered by AI tools that do much of that work automatically. If the learning pathways that created today's experienced engineers are being bypassed, the pipeline of experienced engineers a decade from now may look very different.

However, the talent strategy response to this concern is strikingly thin. Only 4% of respondents cited supporting junior engineers' development in an AI-assisted world as the primary way AI is affecting their approach to engineering talent. The gap between concern and action is significant. Leaders who recognize the risk but have not yet addressed it are building a problem that will compound over time.

The dominant talent response to AI is upskilling. Just shy of half (49%) say upskilling existing engineers to work effectively with AI is the primary way AI is affecting their talent strategy.

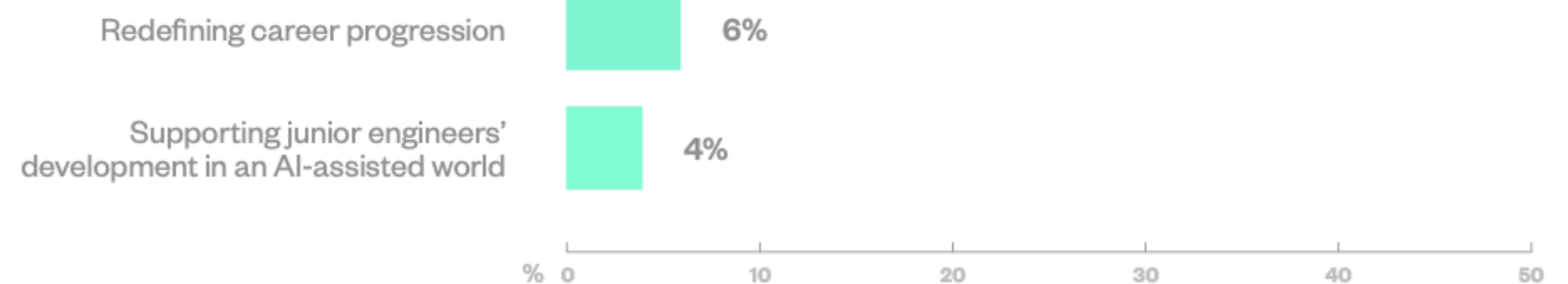
Meanwhile, 18% are increasing **productivity expectations** from teams and 12% are rethinking **hiring profiles and required skills**.

These are reasonable responses to the immediate demands of AI adoption, but they are insufficient to address longer-term questions about career pathways, skill development, and the sustainability of the engineering profession as a whole. Our

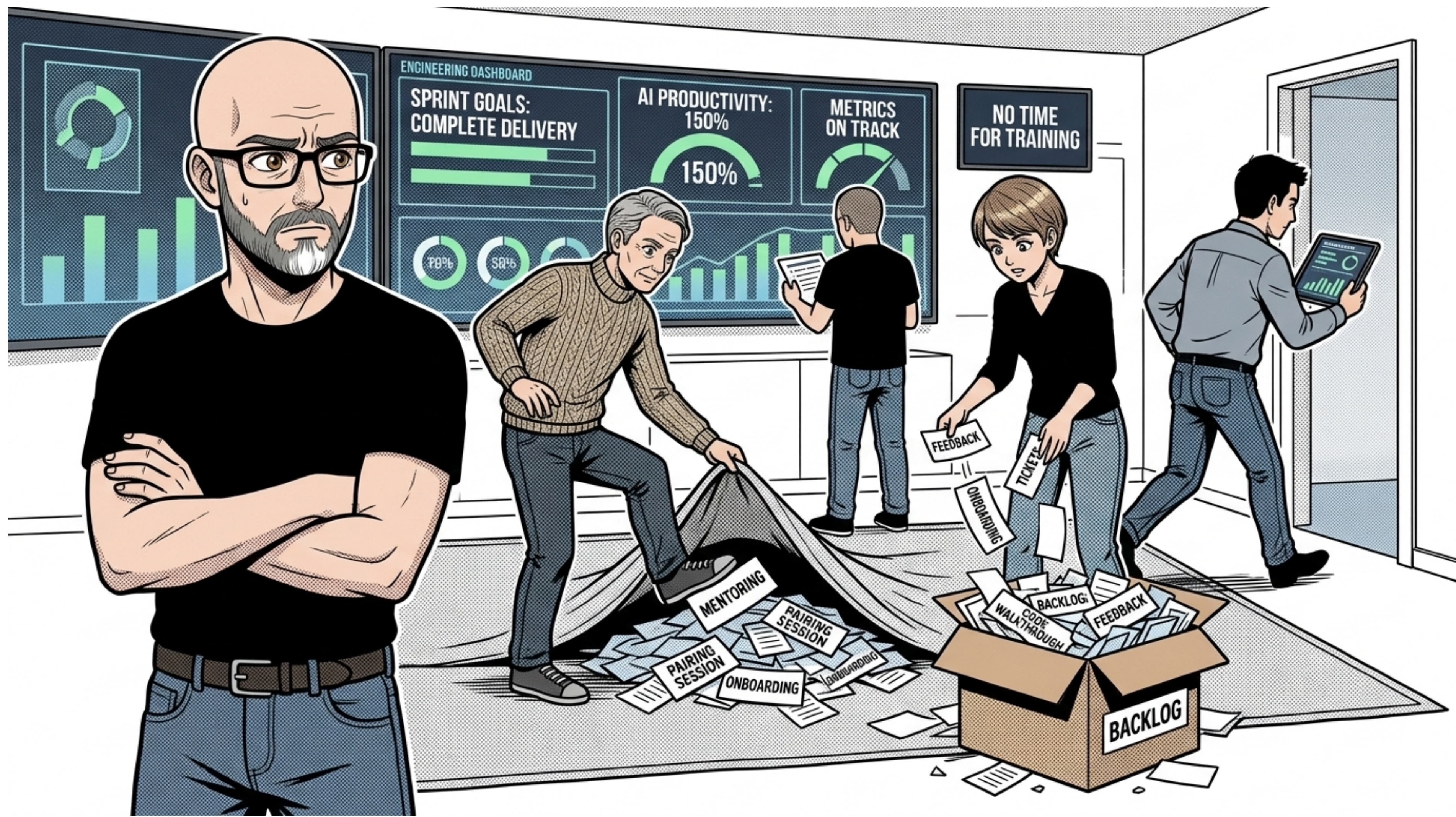
disruption narrative suggests that, say AI reduced their headcount by 2025, far lower than the scale of adoption story might imply. However, 21% believe AI will reduce headcount by 2026, suggesting a meaningful expectation that the impact will accelerate even if it has been limited to date.

For engineering leaders managing team anxiety about job security, the distinction between current reality and anticipated future is important to communicate clearly.

Only
4% of respondents cited supporting junior engineers' development in an AI-assisted world



80%

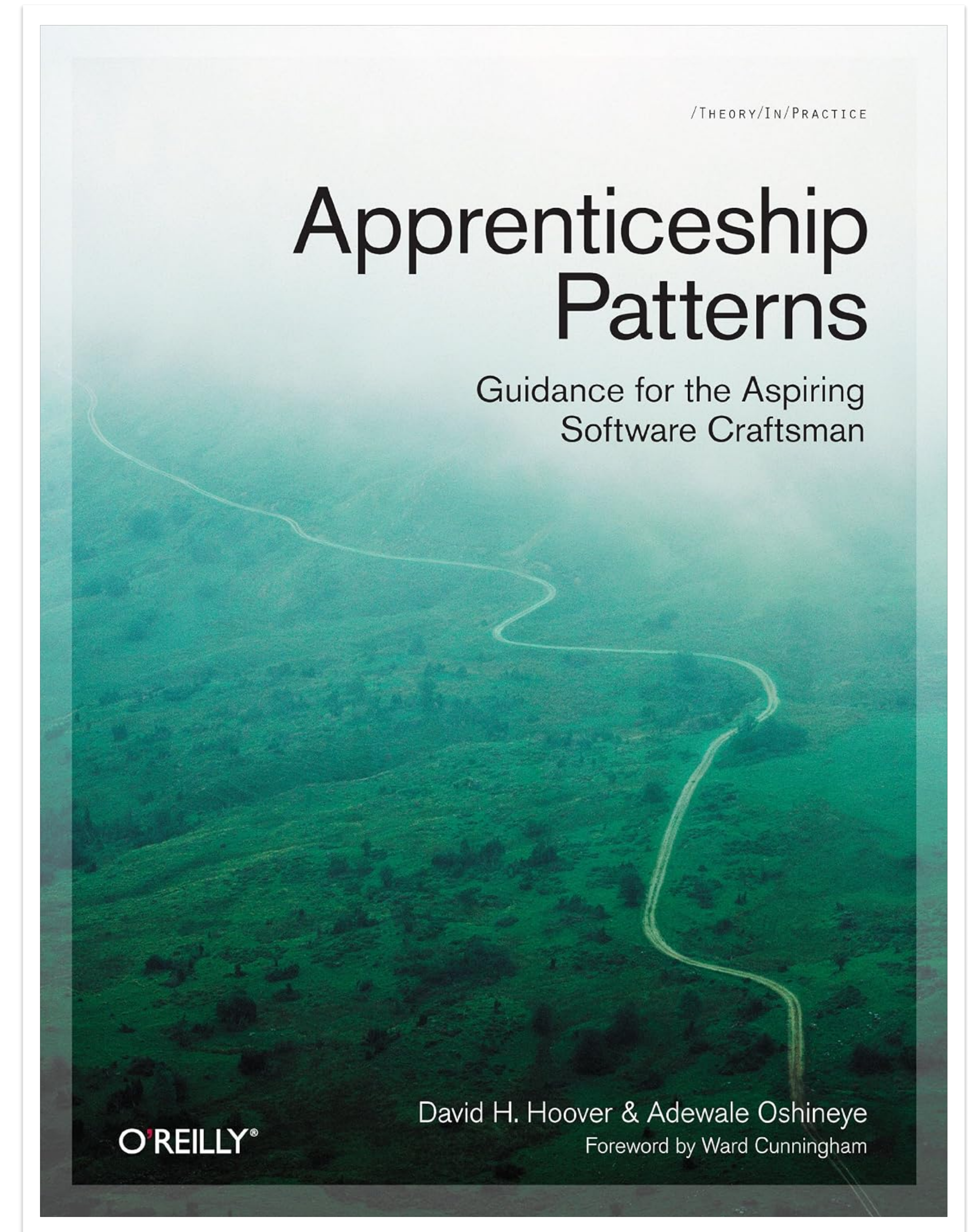


ACT TWO

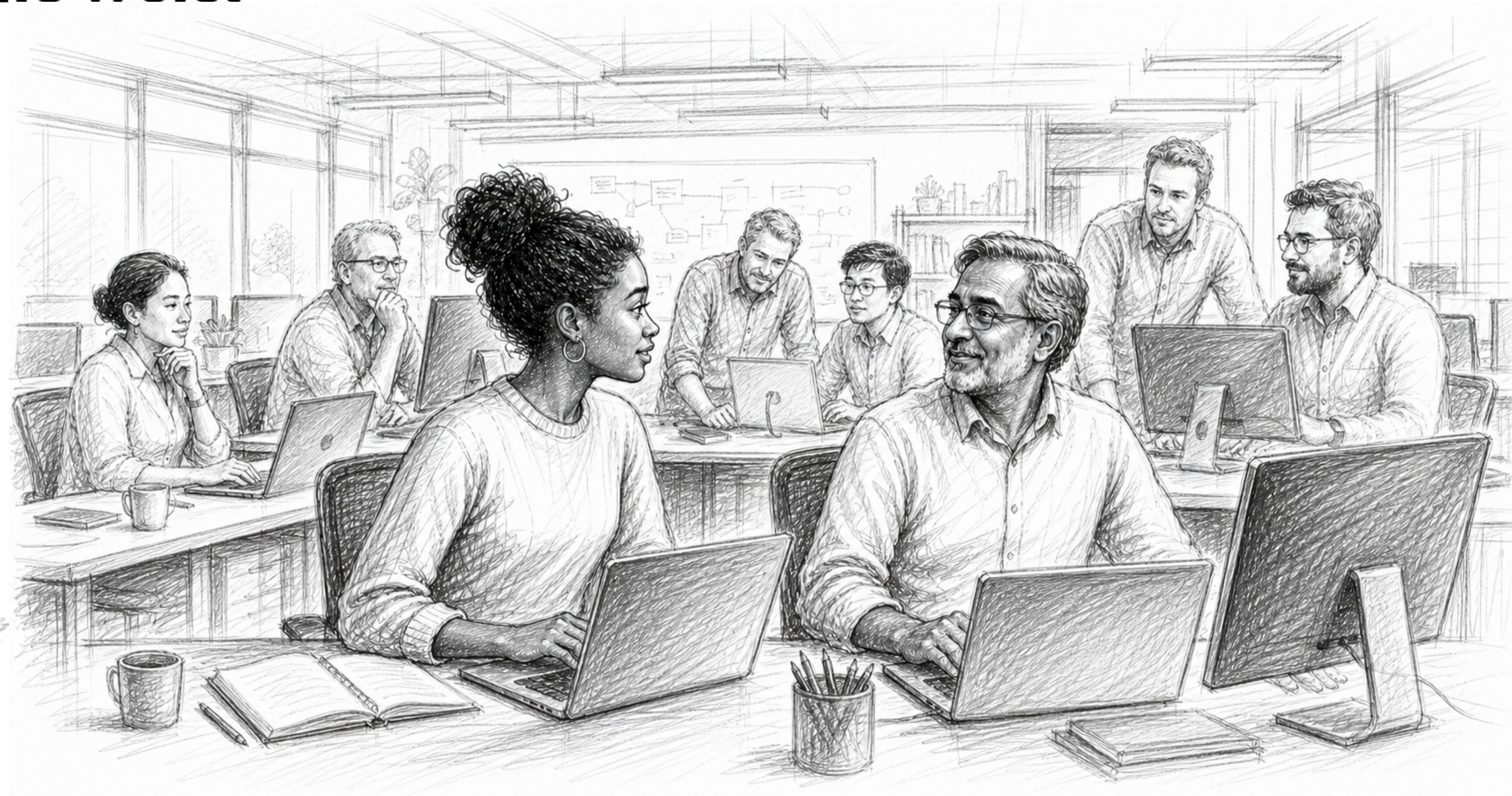
THE **BROKEN** LADDER



The Ladder had a manual



Be the Worst



Put yourself in a **room** where everyone is better than you and absorb as much as you can.

Breakable Toys



Build small, private systems where failure is safe.

Sweep the Floor



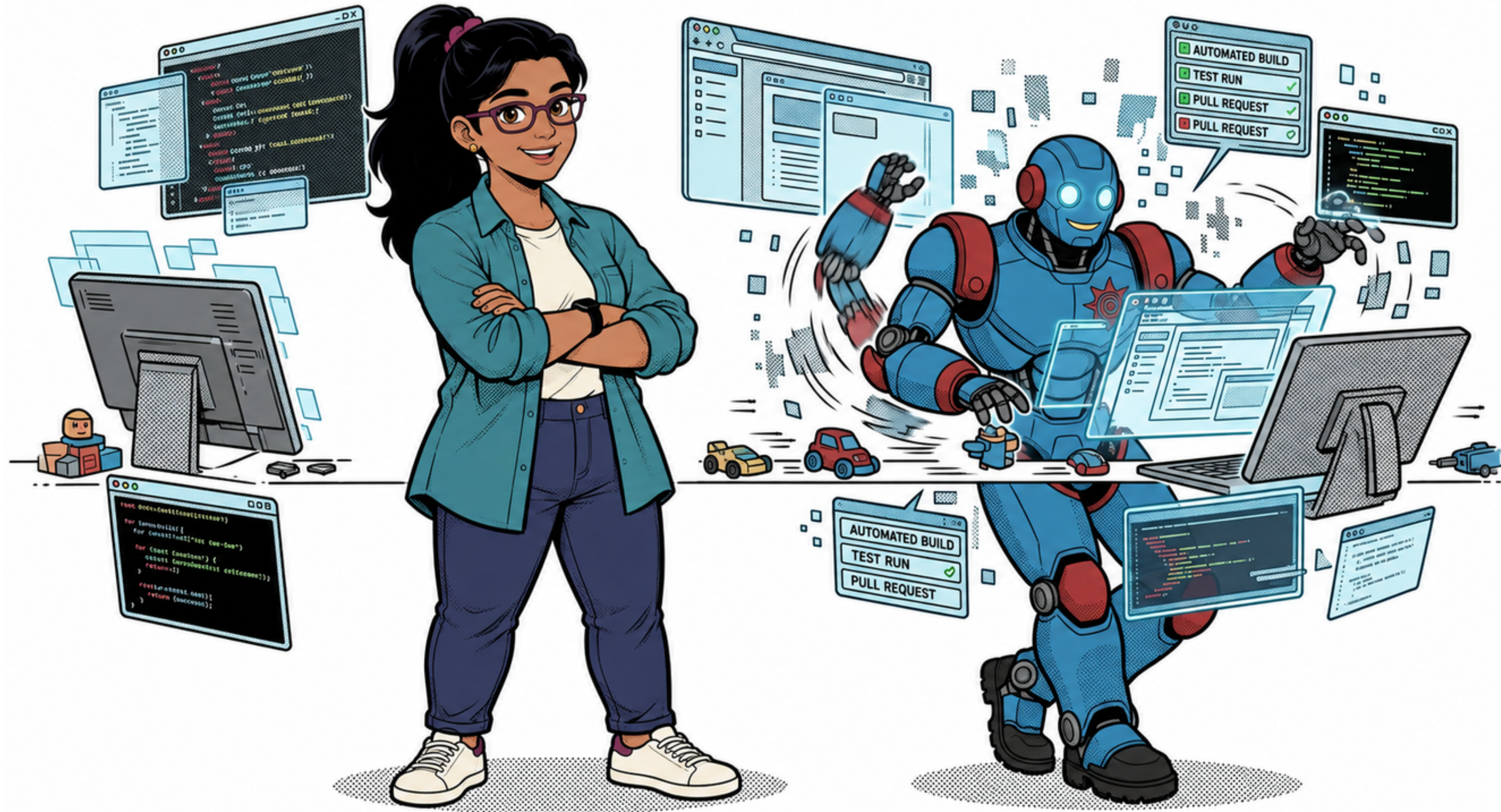
Volunteer for the humble tasks that nobody wants to do.

WHERE ARE WE **NOW**?

Be the Worst: you are alone in the room.



Breakable Toys: the agent fills in the answer before you get the chance to fail.



Sweep the Floor. We bought a Roomba.



Air France Flight 447, travelling from Rio de Janeiro to Paris, with 228 people on board.



Ice crystals block the speed sensors, causing Autopilot to disconnect.



The captain was resting while the junior was driving.



The captain was resting while the junior was driving.





Three minutes. 228 on board.
No survivors.

Captain flying log of the previous 6 months

346

Hours of flying

2

Hours of manual flying

“The training regime was not designed to compensate for a lack of manual high-altitude flying skills”
Air France 447 Accident report



AIRLINES **NOW** REQUIRE **MANUAL** FLYING TRAINING

FAA Advisory Circular AC 120-123: Flightpath Management.

3.3.3.2

Required Maneuvers. Required manual flight maneuvers are contained within the approved part [121](#) training program requirements. This includes requirements such as:

1. Manually controlled slow flight (refer to § [121.423\(b\)](#) and AC [120-111](#), Upset Prevention and Recovery Training).
2. Manually controlled loss of reliable airspeed (refer to § 121.423(b) and AC 120-111).
3. Manually controlled instrument departure and arrival (refer to § 121.423(b) and AC [120-114](#), Pilot Training and Checking (14 CFR Part 121 Subparts N and O, Including Appendices E and F)).
4. Upset recovery maneuvers (refer to § 121.423(b) and AC 120-111).
5. Recovery from bounced landing (refer to § 121.423(b) and AC 120-114).

ACT THREE

ADD **FRICTION** ON PURPOSE



**That Tuesday I had a problem to solve: be in
that 4%**

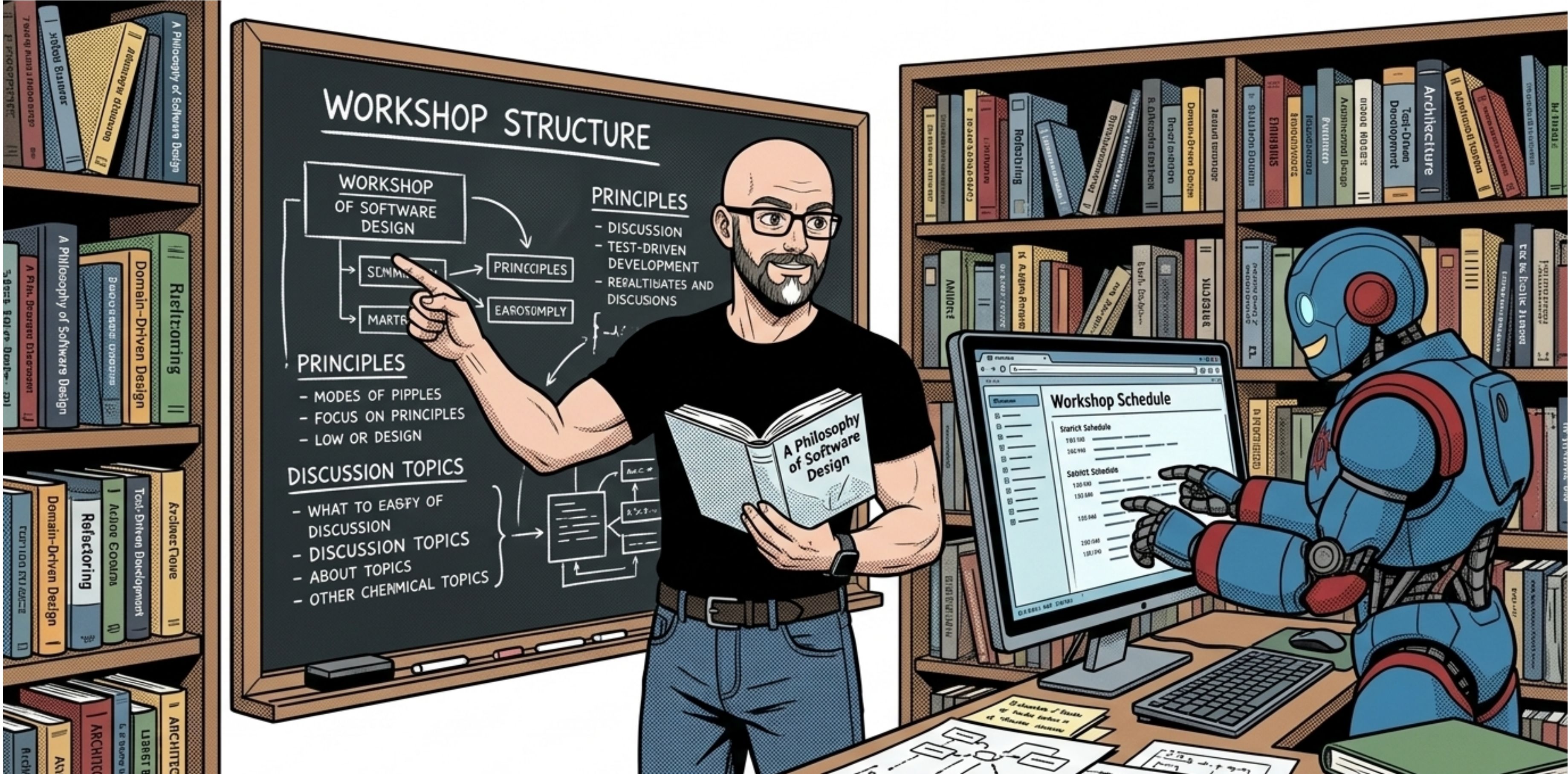
84%

of leaders concerned about junior
skill formation

4%

doing anything about it

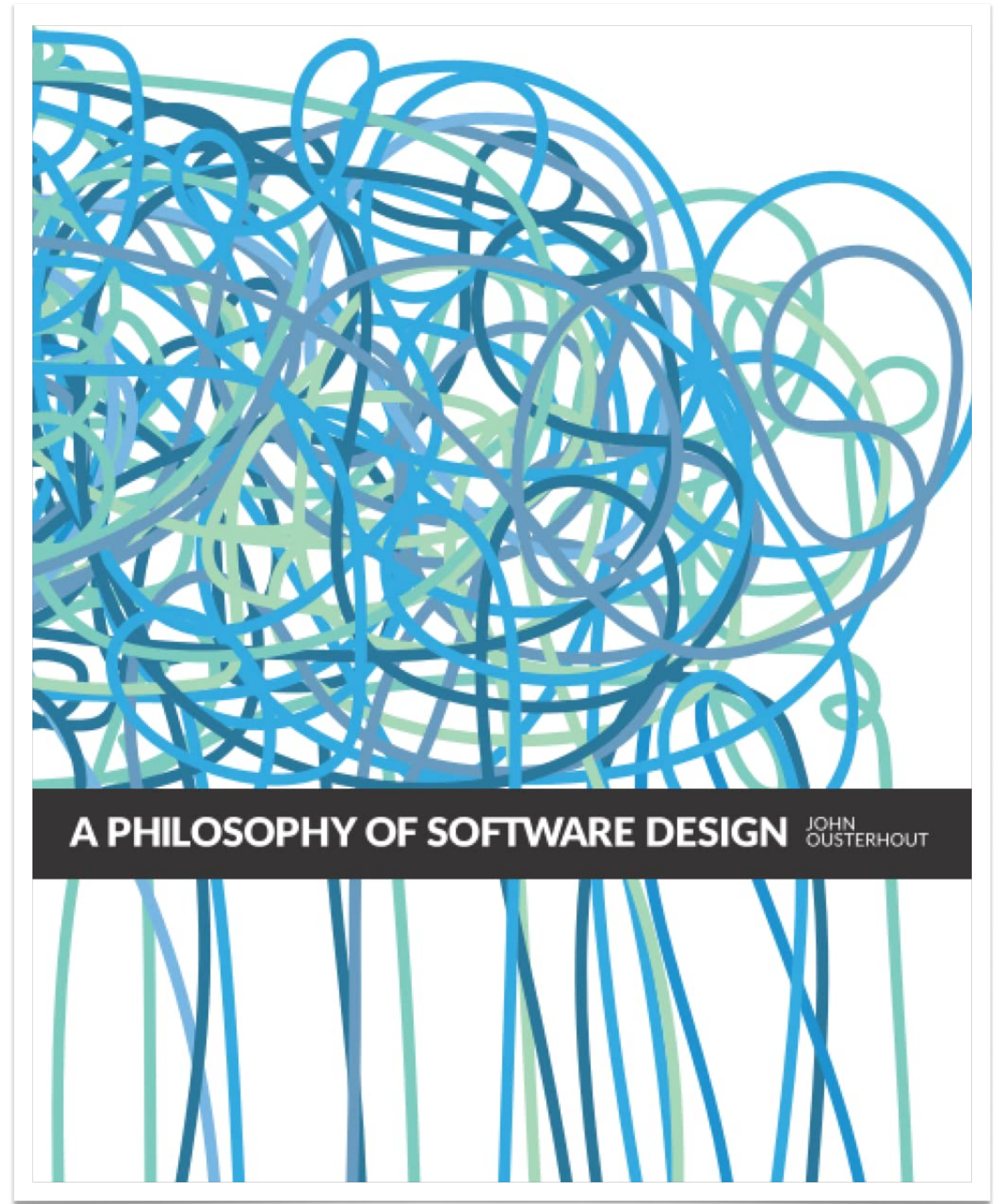
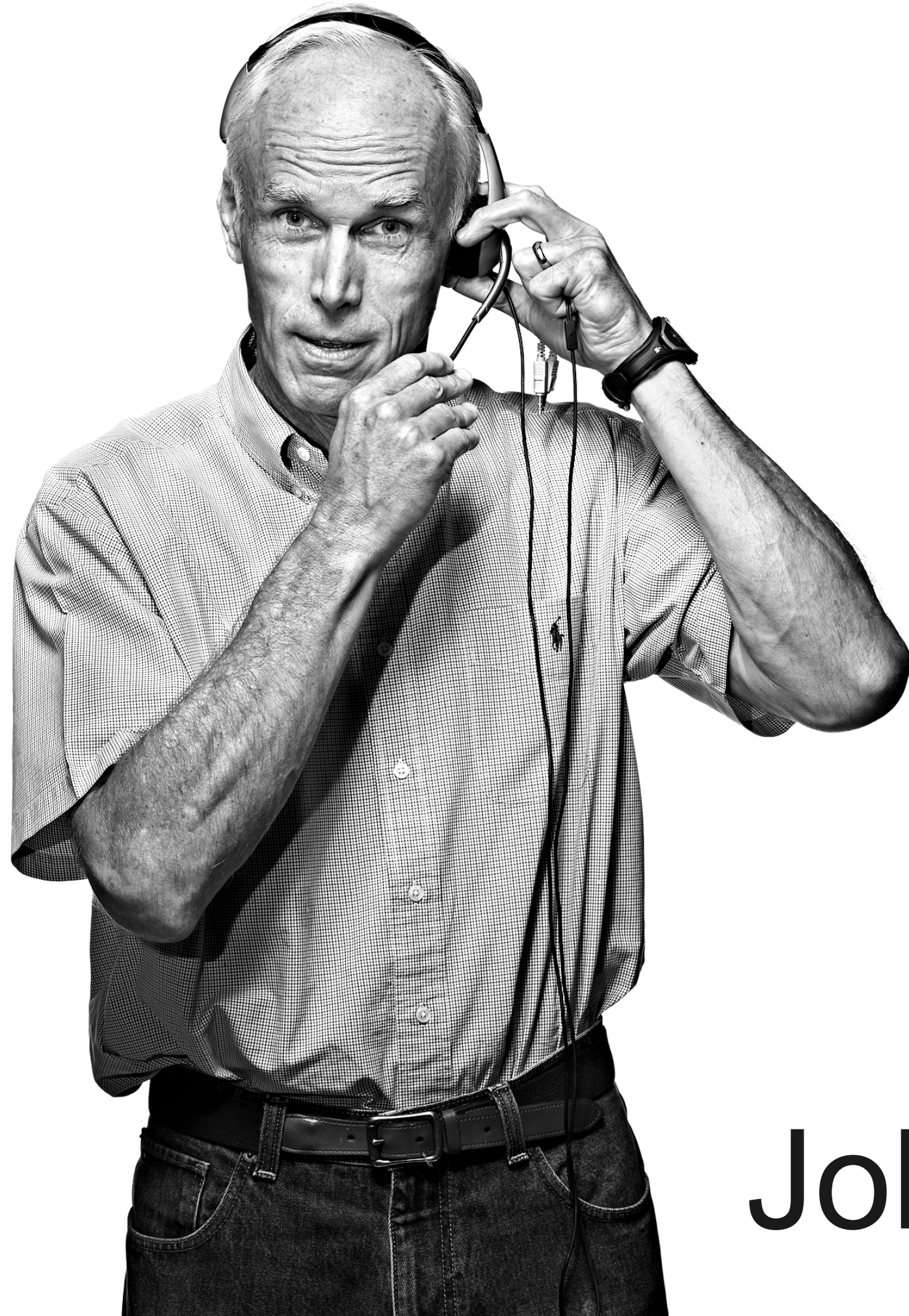
AI SOFTWARE DESIGN WORKSHOP



FORMAT:

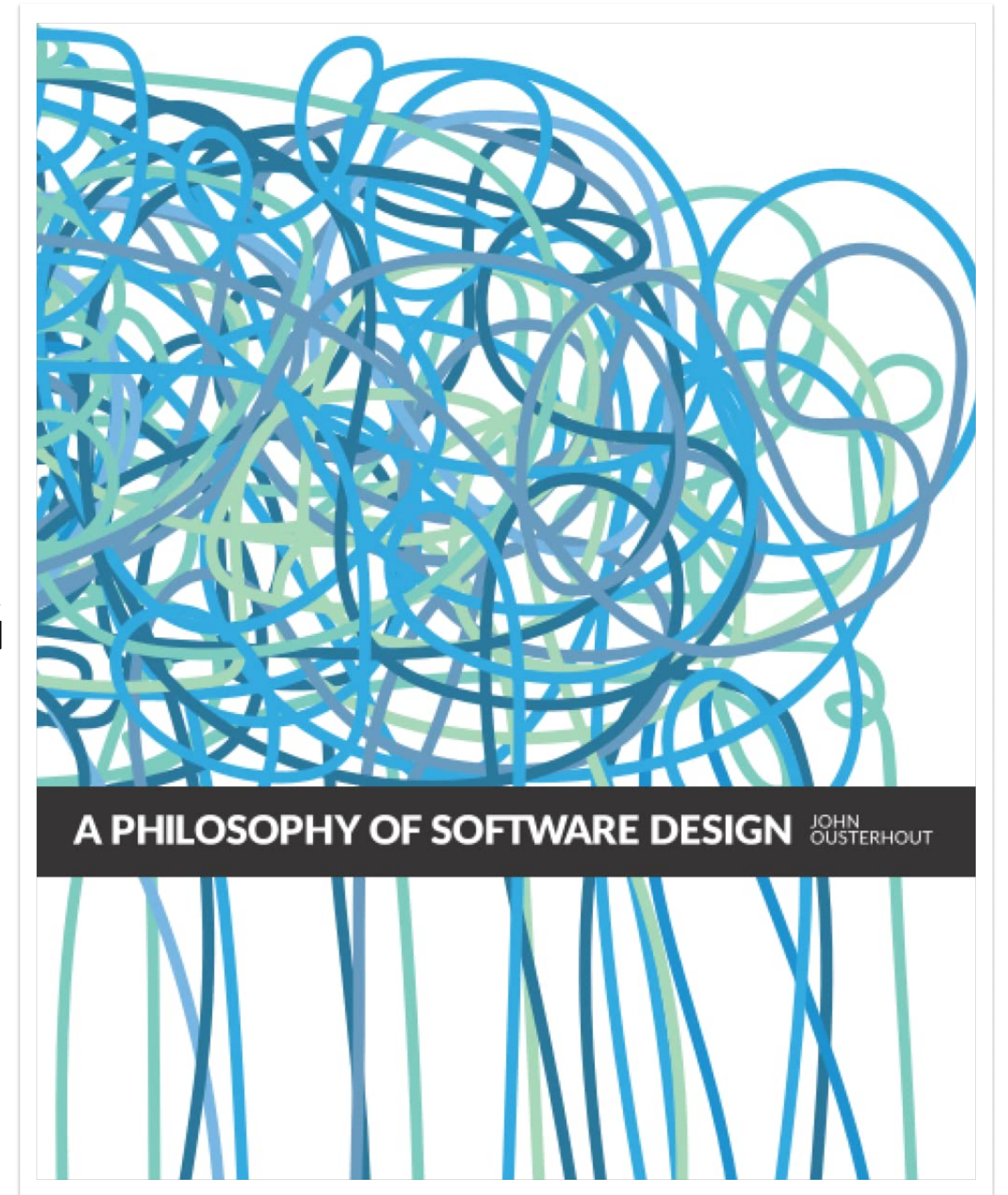
- IN **PERSON**
- BACK TO **BASICS**
- ADD **FRICTION**

Back to basics.



John Ousterhout

DEFINE **COMPLEXITY** IN SOFTWARE ENGINEERING



EXERCISE **ONE**:

AI OFF SESSIONS



CONTEXT

Engineers turn to the agent before forming their own view.



ACTION

Pick a genuine, ticket-sized problem from your own codebase.

15 minutes to design a solution.



A “Mark all as read” button on a notifications screen.

Close the AI. *Not the laptop.* Fifteen minutes. Prose, not code.

- 1 What does it do, **in one sentence**, for the user?
- 2 Which **states** matter? Empty, loading, failed, offline.
- 3 Which **edge case** bites first?
- 4 What am I **least sure about**? Blank is fine: write “I don't know yet.”

15 : 00

Start

Reset

length 5 · 10 · **15** min

Then: read your first sentence aloud, and open the AI. Now there are two designs in the room.

Here is my design note for a "Mark all as read" button on a notifications screen:

[paste your note]

Criticise it against three criteria:

- change amplification (one change forces many edits)
- cognitive load (too much to hold in your head at once)
- unknown unknowns (runtime surprises invisible in the code)

Tell me:

1. What in my note do you agree with, and why?
2. What is wrong or overstated?
3. What did I miss that matters?

Then suggest two designs, minimal and pragmatic. Prose only, no code.
Recommend one and say what you would reject from the other.

When it replies, answer out loud: What did you catch that AI didn't? What did AI catch that you missed?

Then write one sentence: I am picking X because Y. I am rejecting Z because W.

DISCUSS

- What did the AI capture that you didn't?
- What did you do that the AI didn't?



EXERCISE **TWO**:

THE **CODE SMELLS** QUIZ



CONTEXT

AI can generate more code than we are able to review.



DEFINITION

“A **code smell** is a **surface** indication that usually corresponds to a **deeper problem** in the system.”

Martin Fowler



A small banner in a subscription app, shown when the free trial is nearly over.

```
struct TrialEndingBanner: View {
    let trial: TrialStatus

    var body: some View {
        if trial.daysRemaining <= 3 {
            Label("Your free trial ends in \(trial.daysRemaining) days",
                  systemImage: "clock.badge.exclamationmark")
                .font(.subheadline)
                .padding(12)
                .background(.yellow.opacity(0.2),
                           in: RoundedRectangle(cornerRadius: 8))
        }
    }
}
```

Justified, or a smell — which?

Justified — earns its keep

Change amplification

Information leakage

Shallow module

Different layer, different abstraction

A small banner in a subscription app, shown when the free trial is nearly over.

```
struct TrialEndingBanner: View {
    let trial: TrialStatus

    var body: some View {
        if trial.daysRemaining <= 3 {
            Label("Your free trial ends in \(trial.daysRemaining) days",
                systemImage: "clock.badge.exclamationmark")
                .font(.subheadline)
                .padding(12)
                .background(.yellow.opacity(0.2),
                    in: RoundedRectangle(cornerRadius: 8))
        }
    }
}
```

Justified, or a smell — which?

Justified — earns its keep

Change amplification

Information leakage

Shallow module

Different layer, different abstraction

Right. Different layer, different abstraction. The layout is fine; the trespasser is `<= 3`. "When does a trial count as nearly over" is product policy: marketing's number, A/B tested next quarter, maybe remote config after that. Here it lives inside view layout, between a `Label` and a corner radius. When the window changes for one experiment, you edit a SwiftUI `body`; when it moves server-side, the view grows a service call. Product policy and pixel layout, two altitudes in one small view. Small views need the boundary too.

BLANK PAGE WAS **BORING**

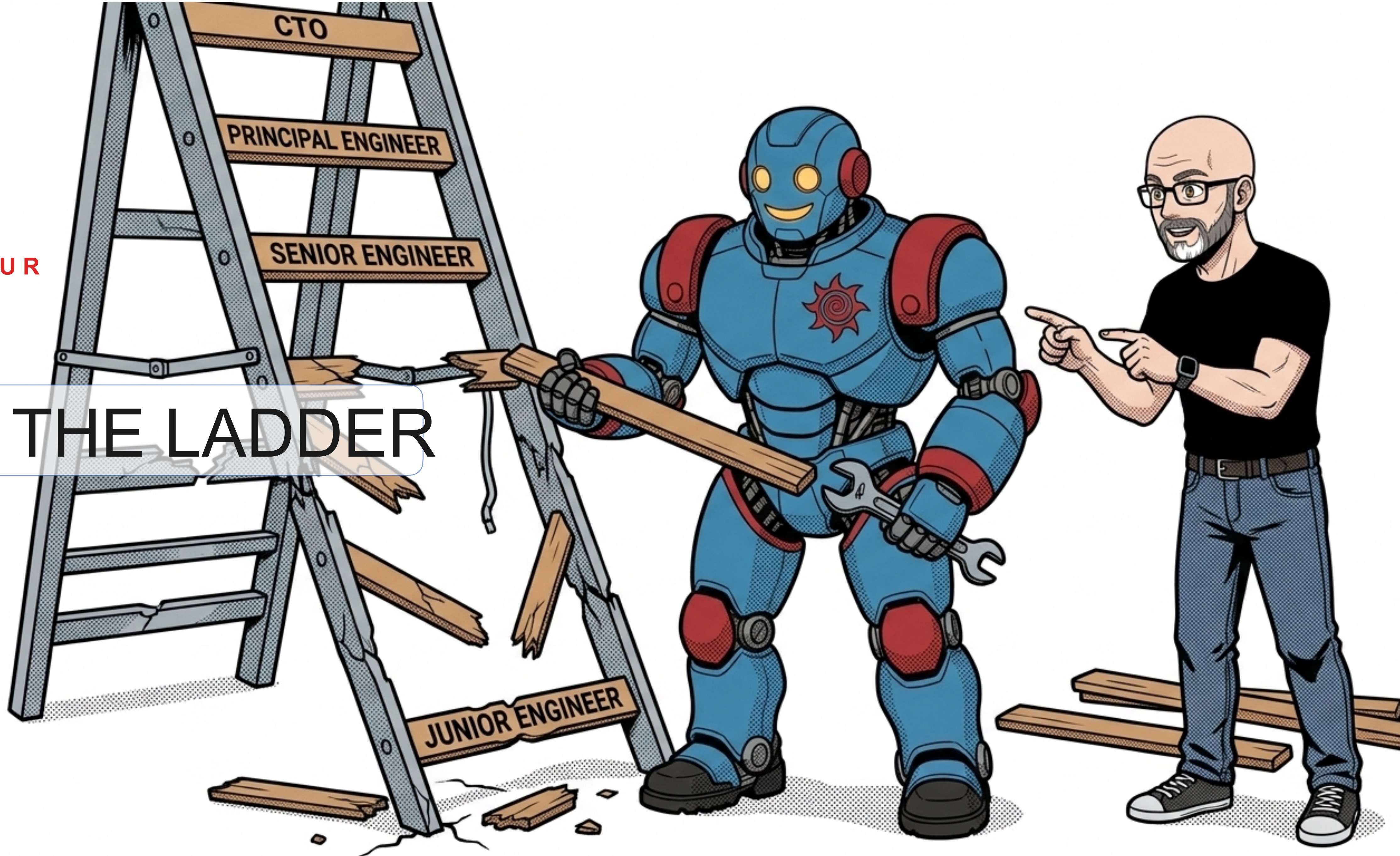
QUIZ WAS **FUN**

IT HAS EVEN FOUND A **REAL BUG**



ACT FOUR

FIX THE LADDER



HYBRID WORK TOOK THE **DESK**

AI TOOK AWAY THE **FRICTION**

AI SOFTWARE DESIGN WORKSHOP:

AI **OFF**

THE **CODE SMELLS** QUIZ

IN PERSON, EVERY WEEK

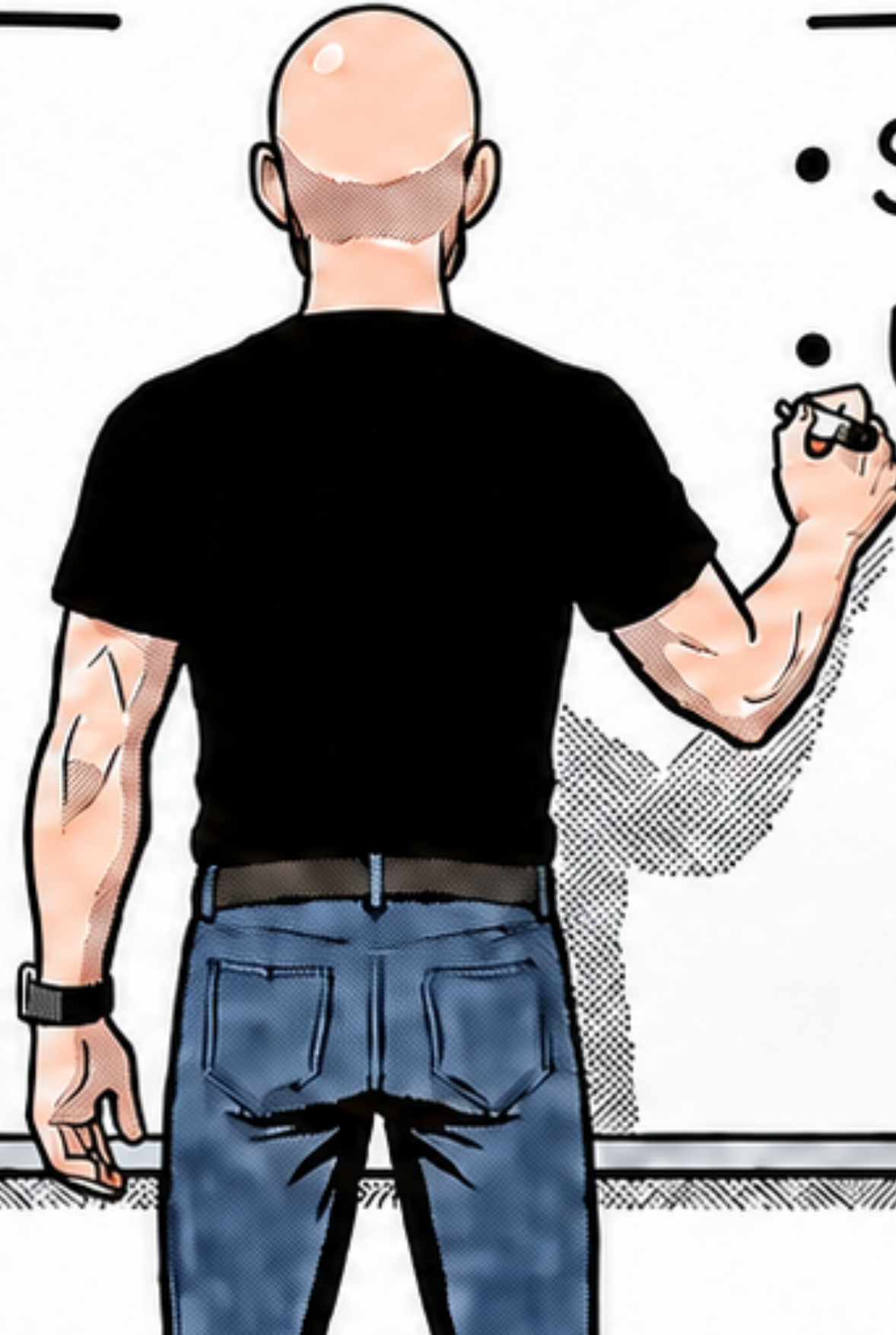
LESSONS **LEARNED**

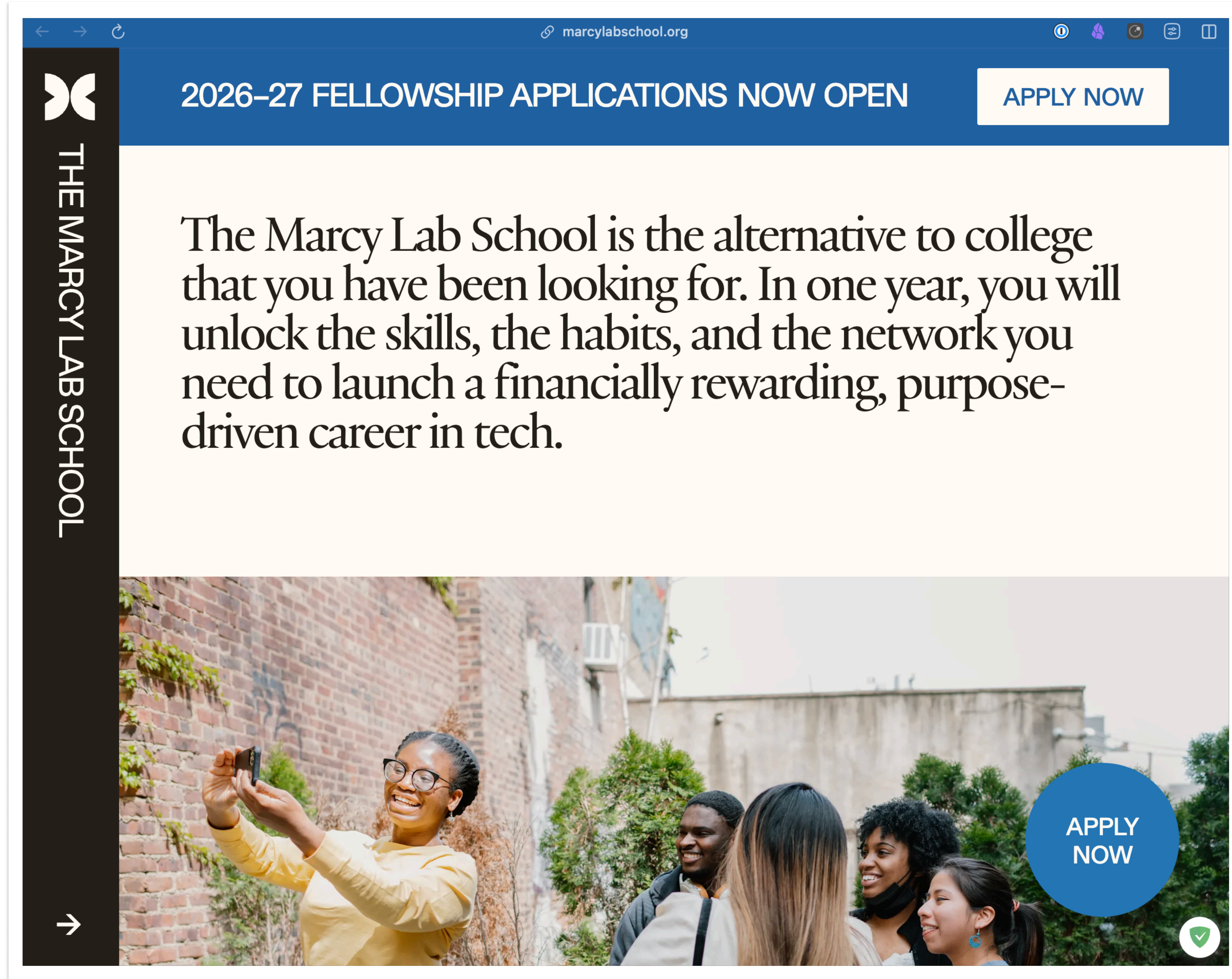
WHAT DIDN'T WORK

- Starting with a blank page → the room froze
- AI as an Oracle

WHAT WORKED

- Show first, then discuss
 - Use real code
- Use AI as a sparring partner





[HTTPS://MARCYLABSCHOOL.ORG](https://marcylabschool.org)

The Stakeholder Hot Seat

AI-Off Sessions

Explain It Without Jargon

Feedback That Lands

Ask Better Questions

Difficult Conversations Lab

The Trade-off Table

Challenge the AI

The Two-Minute Pitch

THERE ARE MORE

Decisions Under Pressure

Explain the Machine

Listen Before You Solve

Requirements Detective

Code Smells Quiz

Break It Challenges

The Prioritisation Game

Negotiate the Scope

Teach It Back

Say No with Options

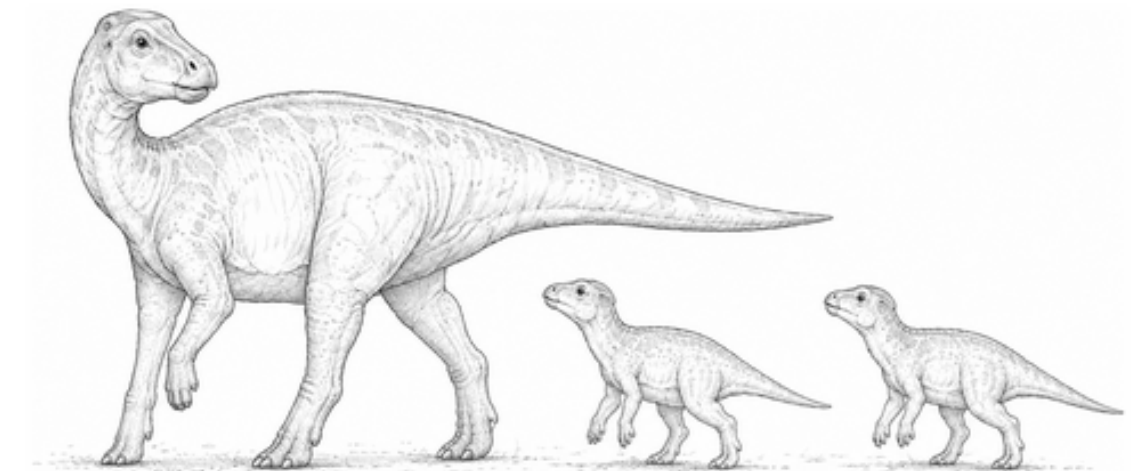
The Incident Room

[HTTPS://GIOSCALZO.COM](https://gioscalzo.com)



FIX THE LADDER

**GROWING JUNIOR ENGINEERS
IN THE AGE OF AI**



GIORDANO SCALZO



GIORDANOSCALZO