

# VIBE CODING

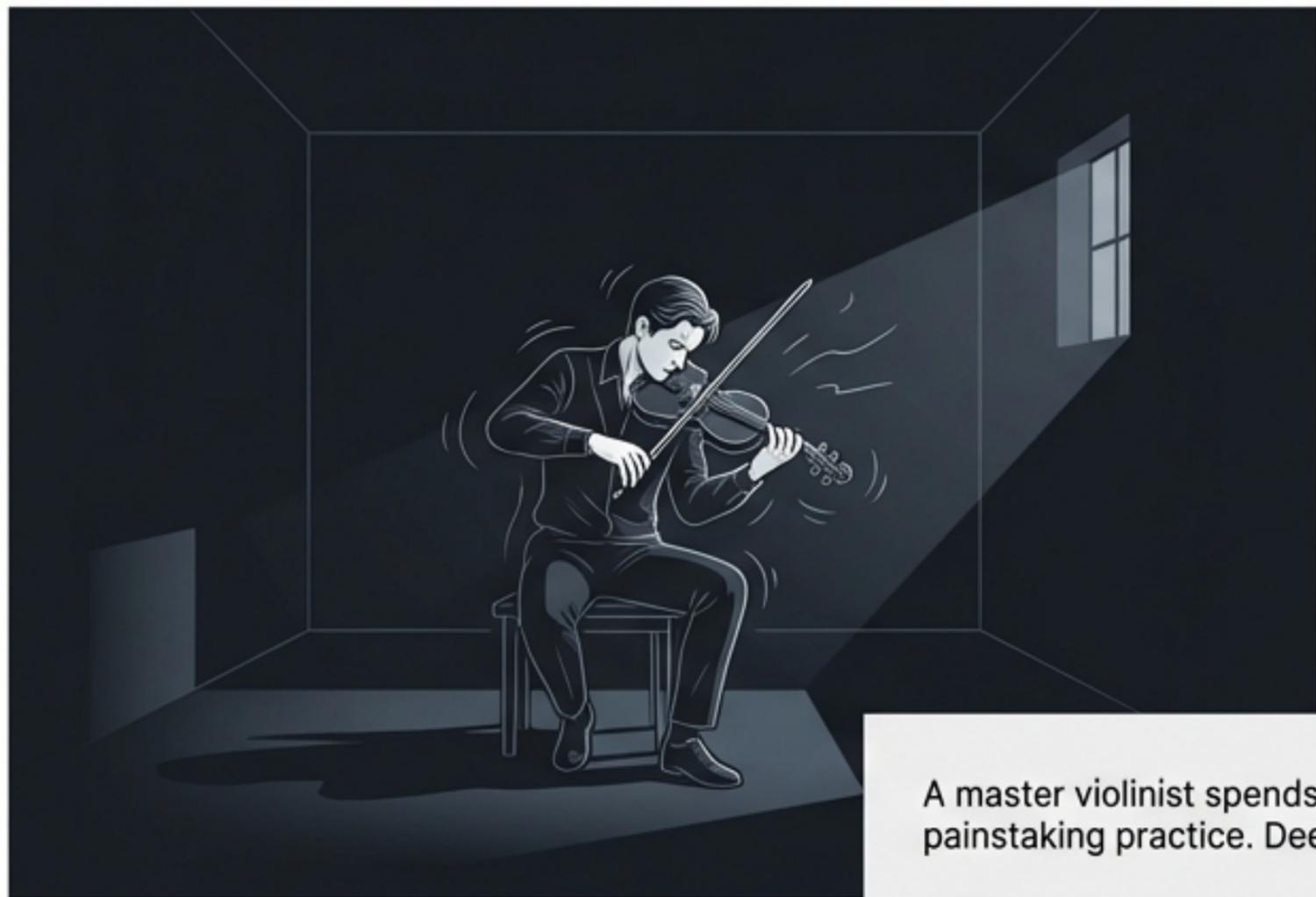


## The Evolution of Intent-Based Software Creation

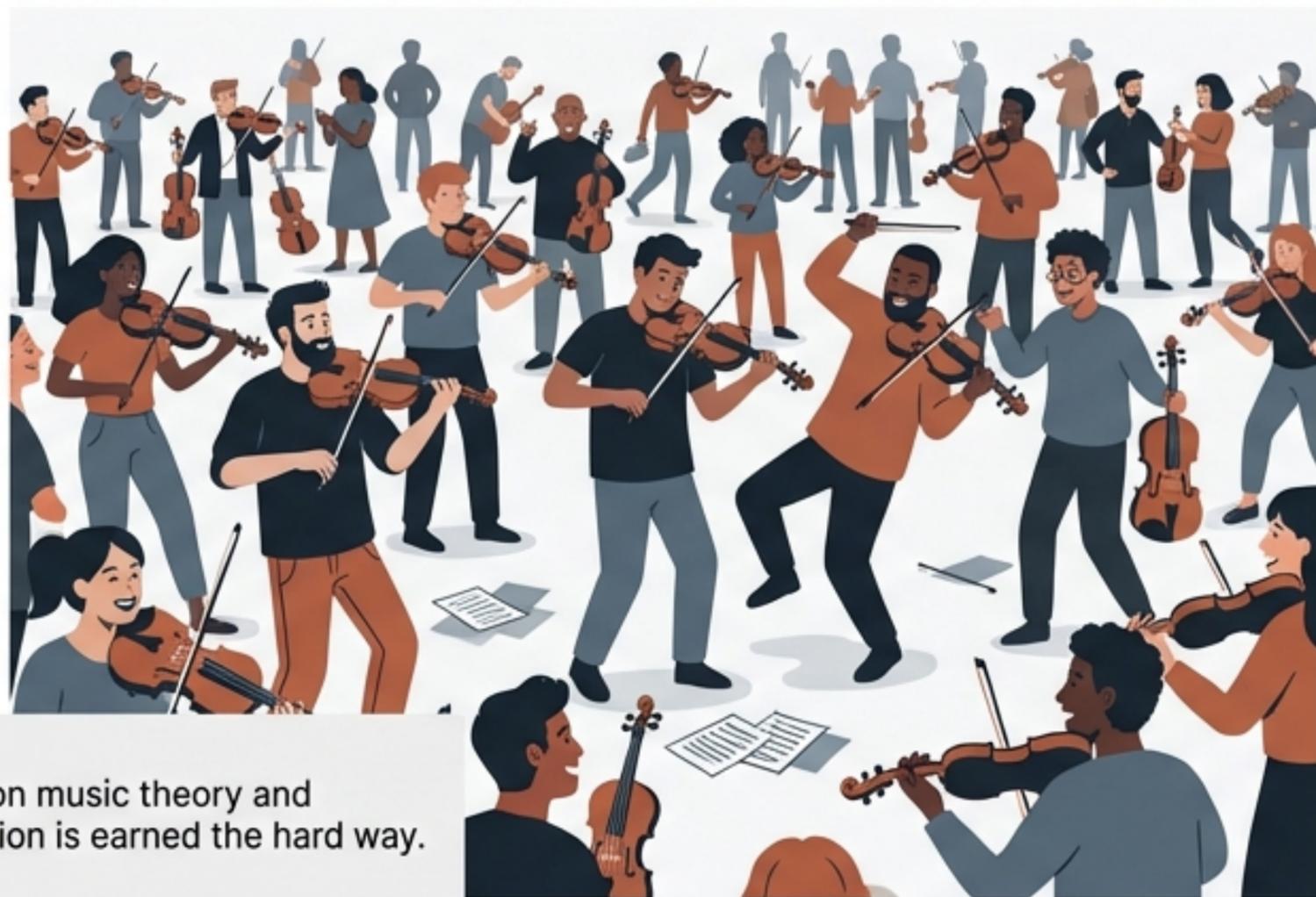
Moving from Syntax to Judgment.

We are witnessing a shift where the unit of work is no longer code, but judgment.  
This deck explores the transition from authoring lines to steering outcomes.

# The Violin Problem



**THE MASTERY OF PAIN**



**THE ACCESSIBILITY OF TOOLS**

A master violinist spends years on music theory and painstaking practice. Deep intuition is earned the hard way.

Suddenly, hundreds of people with no training pick up violins. Most sound awful, but a few sound surprisingly good. The audience can't always tell who put in the years.

That feeling—that something sacred has been diluted—is what developers feel when they see "vibe coders" shipping products without traditional training.

Source: *Computers Were a Mistake (Substack)*

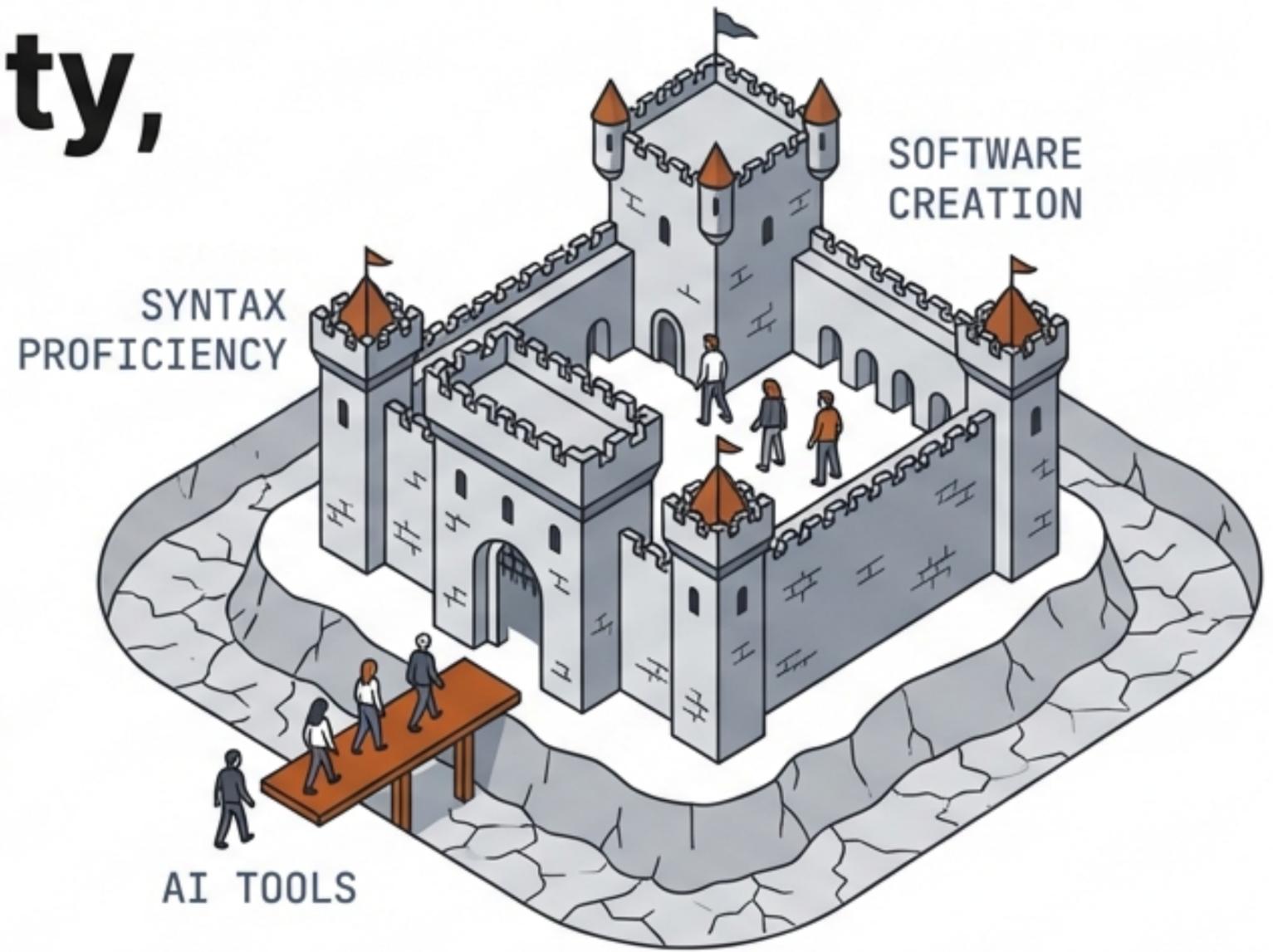
# It Is Not About Quality, It Is About Status

The resistance isn't really about the code being bad. It is about the collapse of a long apprenticeship as the only legitimate path to creation.

For decades, syntax was the moat that protected professional status.

Vibe coding allows people who skipped the suffering to still ship real things.

That feels threatening if you paid the cost and earned your expertise through pain.



**“The tools don’t eliminate mastery.  
They eliminate gatekeeping.”**

# Giving In to the Vibes

This is not a joke. It is a serious reframing of programming when LLMs handle the mechanical work.

- You describe intent rather than writing instructions.
- You steer, refine, and redirect instead of authoring every line.

“There’s a new kind of coding I call ‘vibe coding’, where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.”

— Andrej Karpathy

# Categorically Different Than “AI-Assisted”

## AI-ASSISTED (The Old Way)

- Autocomplete & Snippets
- Boilerplate & Faster Typing
- Focus: Syntax Correctness

**How do I  
write this?**

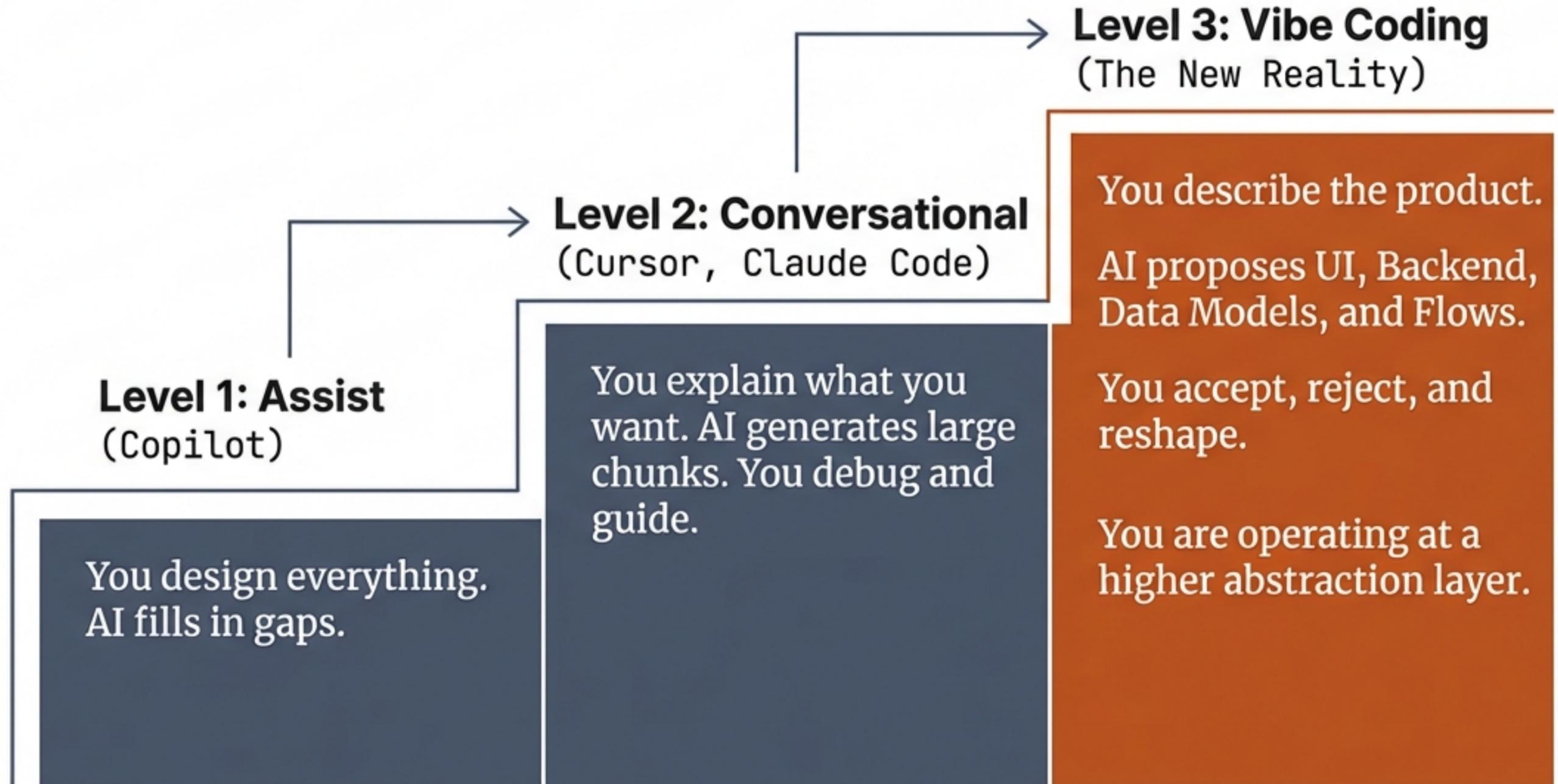
## VIBE CODING (The New Way)

- From syntax to intent
- From implementation to evaluation
- Focus: Direction & Flow

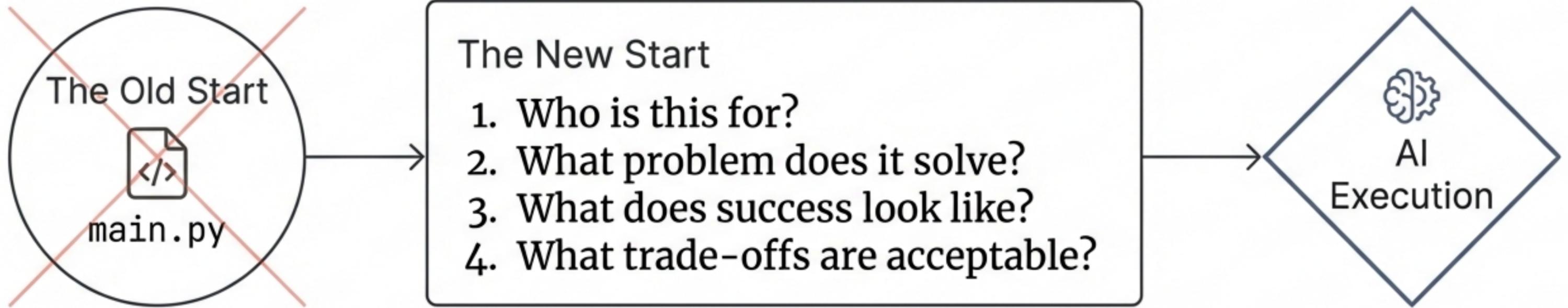
**Is this the right  
thing to build?**

Bottom Line: The center of gravity moves from correctness to direction.

# The Three Levels of Evolution



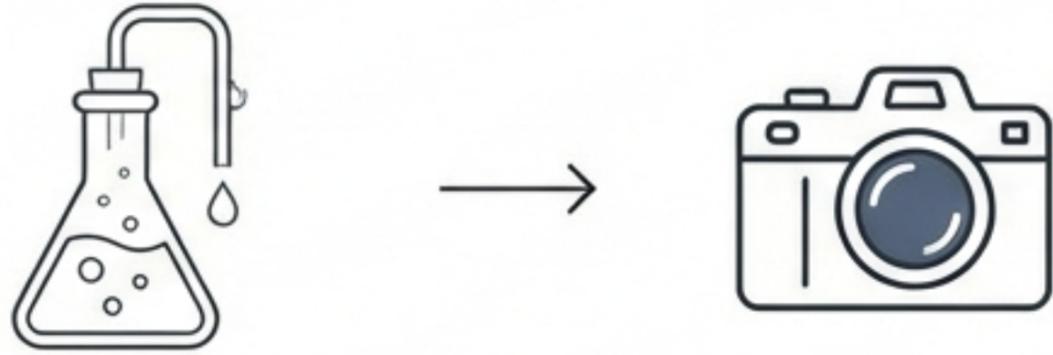
# Clear Thinking Produces Better Software Than Clever Syntax



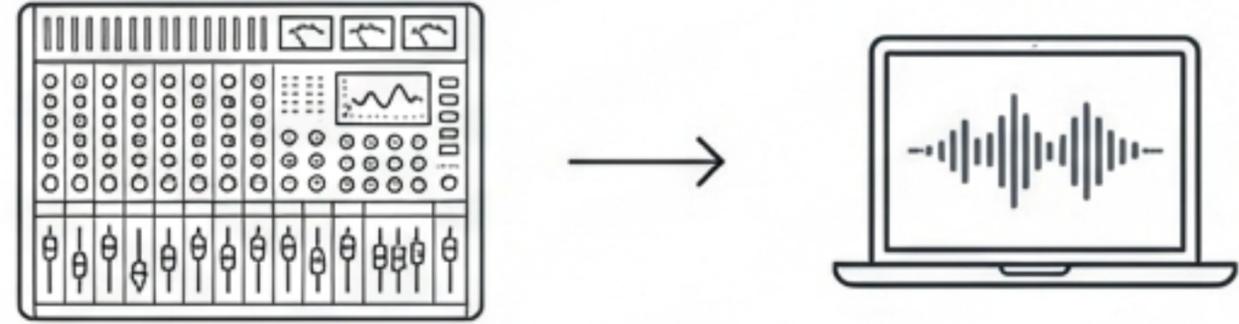
Vibe coding does not start with code files. It starts with definitions of success. When the AI handles the implementation, the human's primary job becomes rigorous clarity of thought.

**AI removes friction, not friction, not responsibility.**

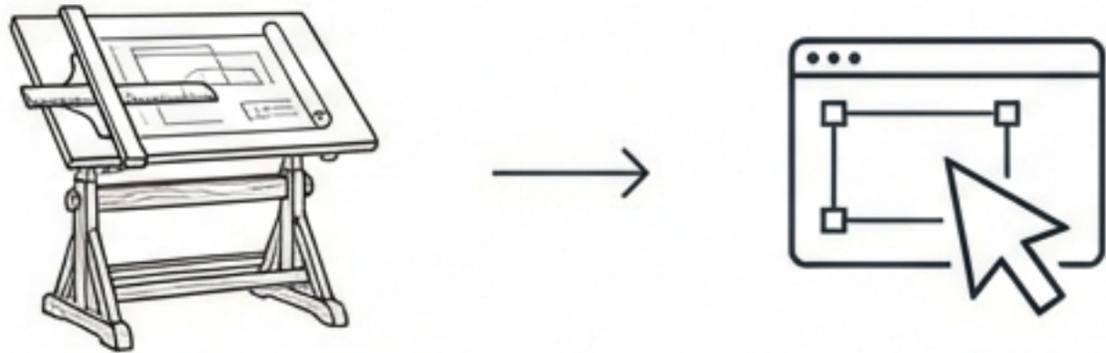
# The Pattern of Democratized Mastery



**Photography:** Digital sensors removed the chemical darkroom barrier.



**Music:** DAWs removed the need for million-dollar studios.



**Design:** Tools like Canva and Figma lowered the floor for visualization.



**Coding:** AI is currently removing the syntax barrier.

In every previous instance, the tool didn't destroy the art form. It expanded the number of people who could participate.

# The Acceleration Curve Is Relentless

2021

It can't even autocomplete a line.

2022

It can't even write a whole function.

2023

It can't even pass a coding interview.

2024

It can't even build an app.

2025

It can't even handle complex projects.

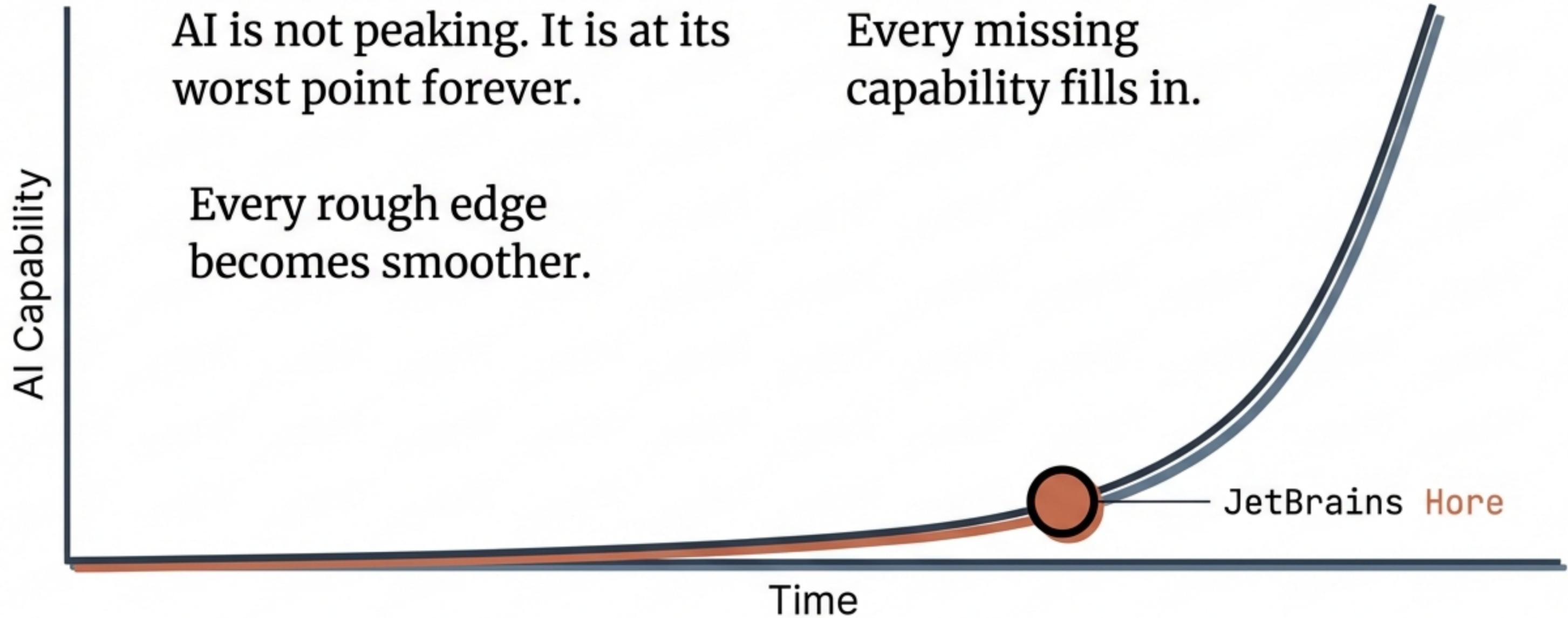
2026

**Oh no.**

This is how  
technological shifts  
look in hindsight:  
slow, then sudden.

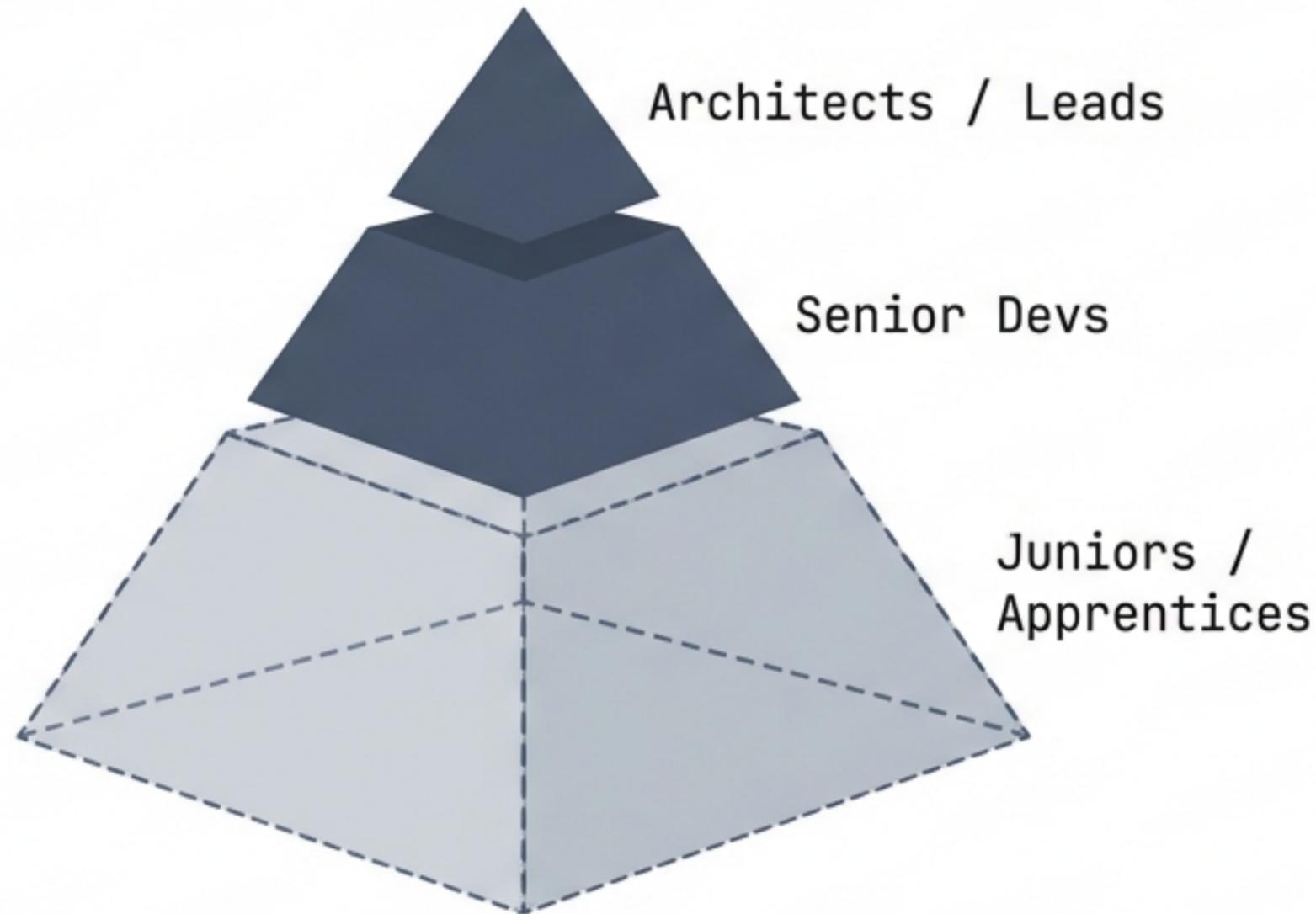
Data: Aaron D. Francis

# The Worst It Will Ever Be



**Waiting for stability is a losing strategy.**

# The Hollow Middle



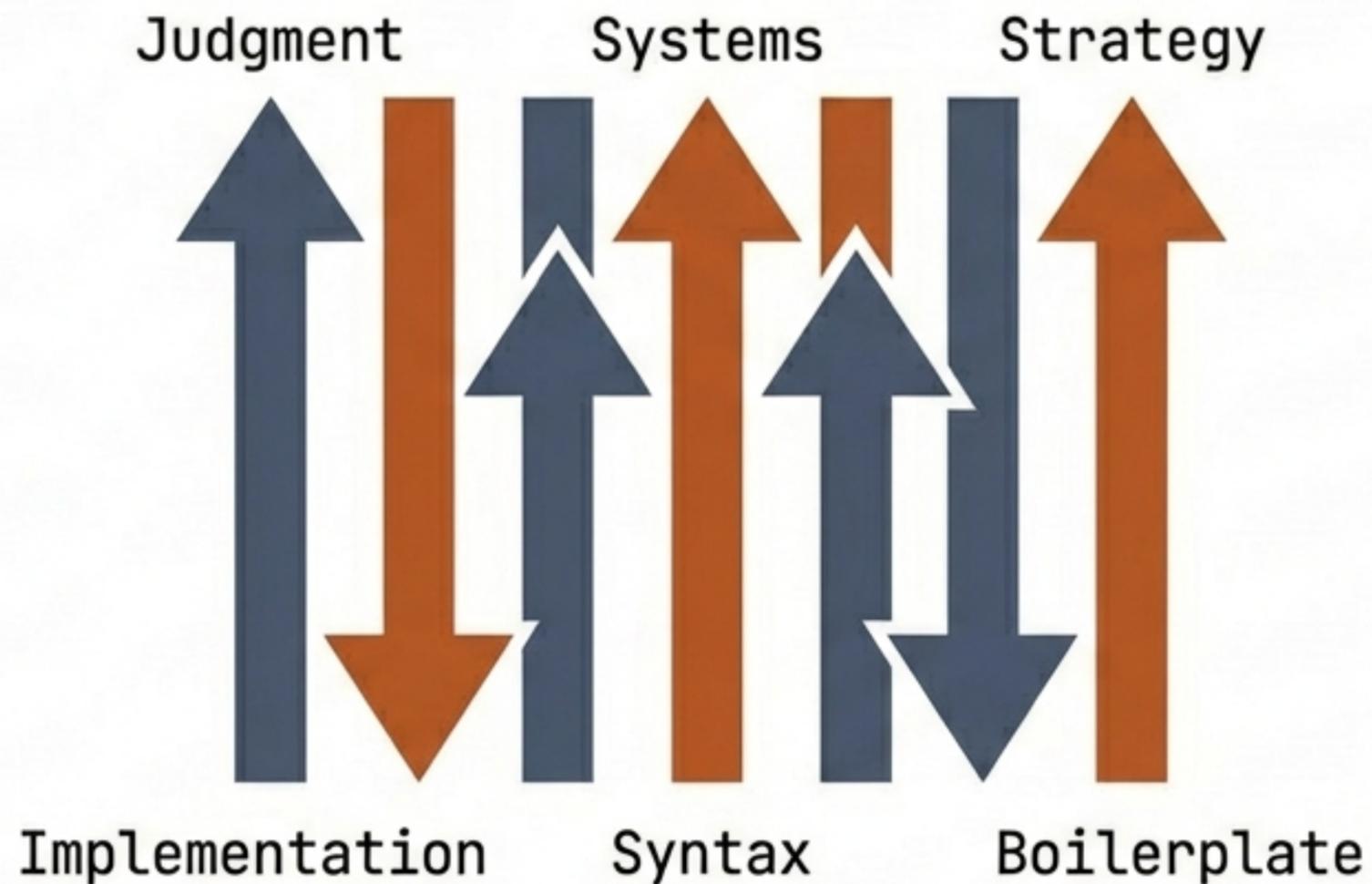
## THE UNCOMFORTABLE TRUTH

AI doesn't eliminate all programming jobs, but it eliminates the bottom rung.

- Fewer junior roles
- Fewer apprenticeship paths
- Harder progression ladders

This creates real anxiety and "this won't work" narratives. But the direction doesn't change.

# Work Rolls Upward



## What AI Cannot Replace

- ✓ Architecture decisions
- ✓ Security thinking & Trade-off analysis
- ✓ Product sense
- ✓ Client communication & Market understanding

These skills become *MORE* valuable as code itself becomes cheaper.  
We move from syntax to systems.

# Should You Learn to Code?

**YES.**

To express ideas precisely, understand constraints, and collaborate effectively with AI.

**BUT...**

Not for job safety, credentials, or protection.

Coding is becoming literacy, not a moat.

# Opportunity vs. Disaster

## For Entrepreneurs

- • Tools are democratized.
- ↳ • Ideas ship faster.
- 🚀 • Leverage explodes.

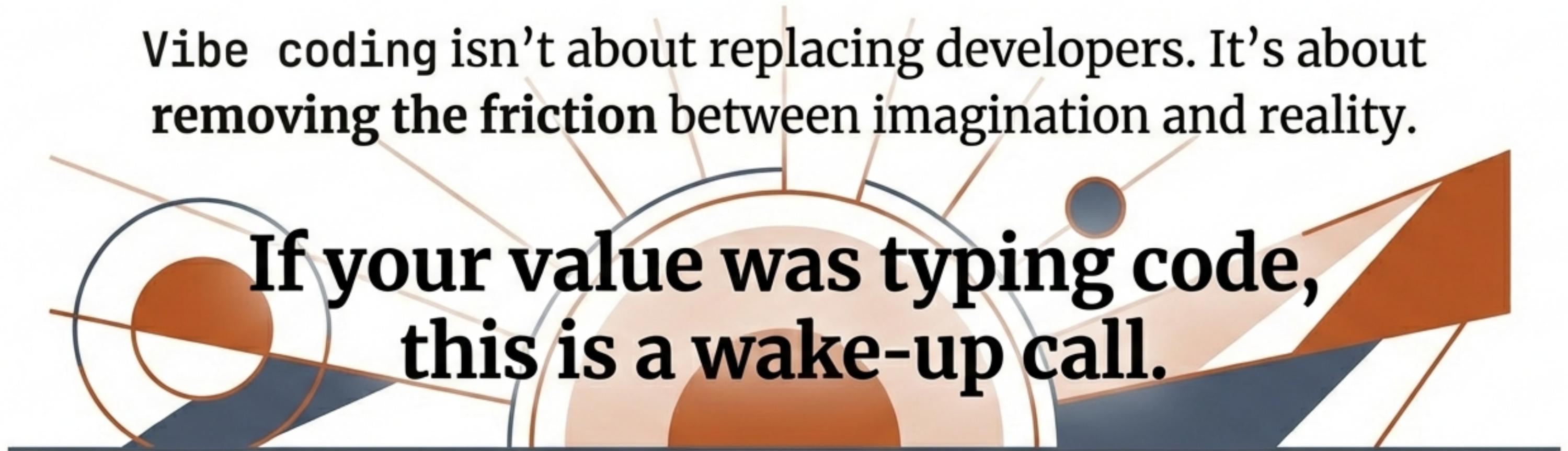
## For Traditional Career Ladders

- • Entry points shrink.
- ↳ • Timelines compress.
- ↳ • Safety disappears.

Same technology.  
Very different outcomes.

# From Writer to Director

Vibe coding isn't about replacing developers. It's about removing the friction between imagination and reality.



**If your value was typing code,  
this is a wake-up call.**

**If you have ideas, this is one of the  
most exciting moments in decades.**

THE END