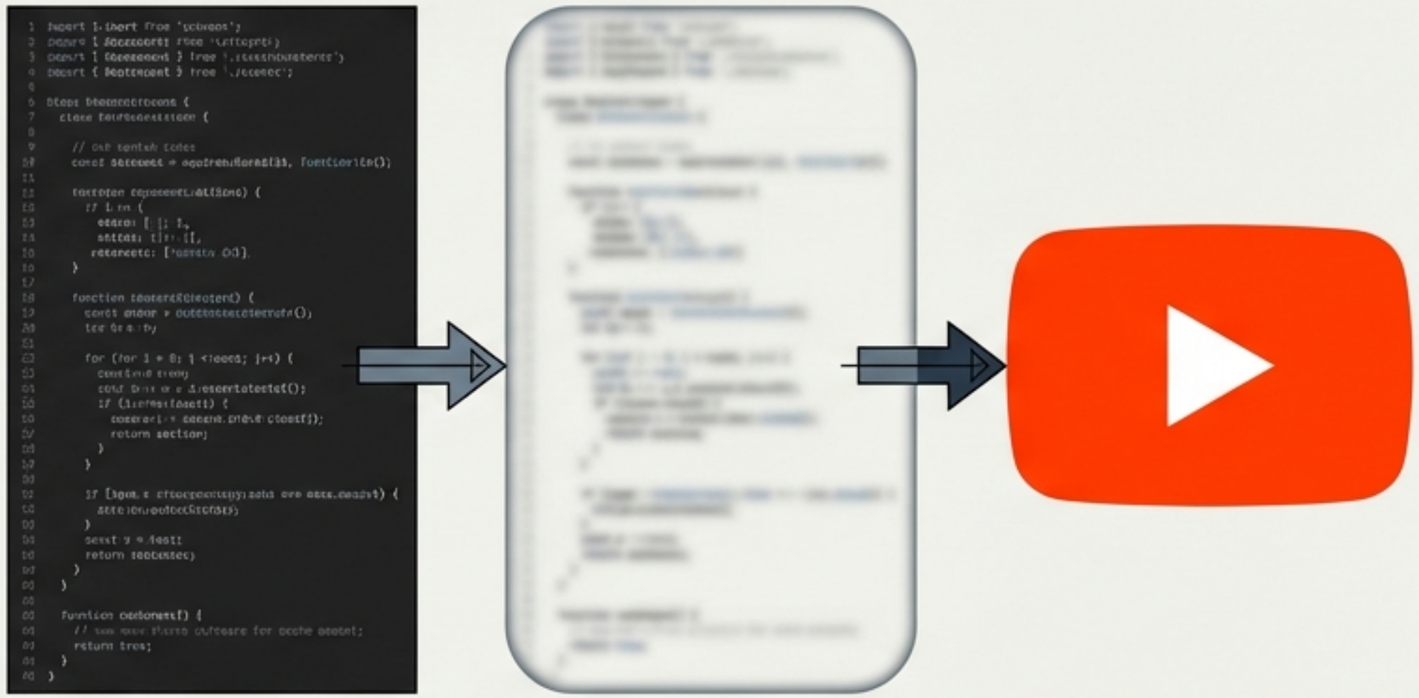
The YouTube Moment for Software

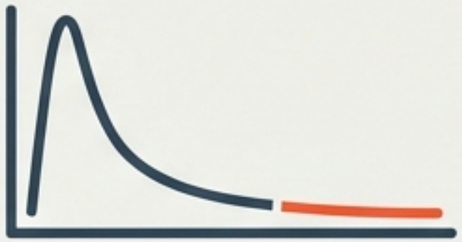
How AI is transforming coding from a technical trade into a mass medium for expression.



INSIGHTS FROM A16Z / ANISH A (2026)

Coding is entering its mass adoption phase

THE TREND



Software creation is undergoing a ‘long-tail’ expansion distinctively similar to video in 2005. Niche, personal, and single-use software is now viable.

THE DRIVER

AI tools have compressed development time from weeks to hours, effectively removing the ‘syntax barrier’ that kept non-engineers out.

THE SHIFT

Software is moving from strictly ‘Utility’ (solving business problems) to ‘Expression’ (creative, funny, niche, cultural).

THE OPPORTUNITY

Unlike video content which decays rapidly, software accrues value over time. We are entering the era of the ‘citizen developer’.

History doesn’t repeat, but the mechanics match

The Hollywood Era

Gatekeepers, Green Lights, Deep Expertise.

The Indie Era (90s)

Lower Budgets, Outsiders (Tarantino/Soderbergh).

The YouTube Era (2000s)

Zero Barriers, Camera in every pocket.

Insight: In 2005, YouTube didn’t seem to fill an obvious gap. Two decades later, it is a $550 billion business more culturally relevant than traditional TV.

We are leaving the ‘Hollywood Era’ of Code

The Enterprise Era

Tens of millions to build. Deep engineering teams required.

The YC / Indie Era

Small teams (Airbnb/Coinbase). Technical chops still required.

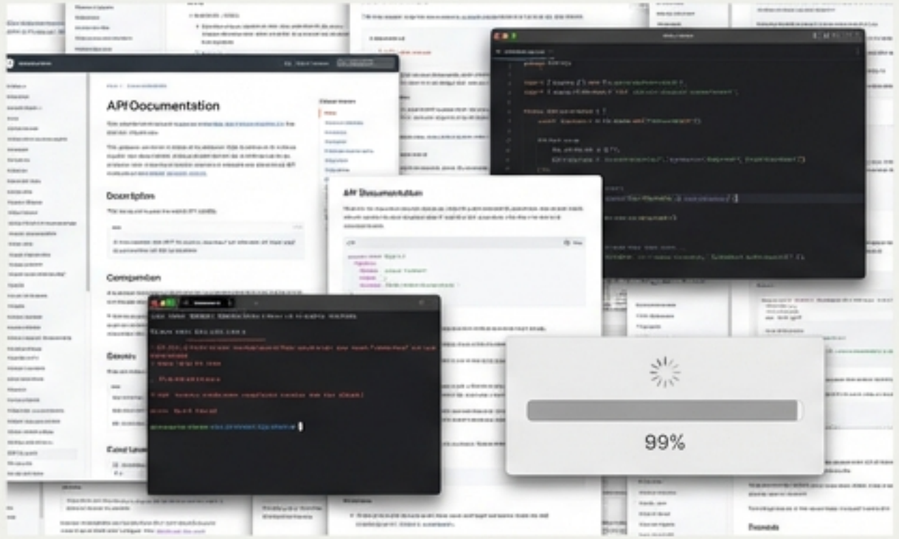
The “YouTube” Era (Now)

LLM at fingertips. No syntax knowledge needed.

If you previously had excuses for not building something, they just got a lot less convincing.

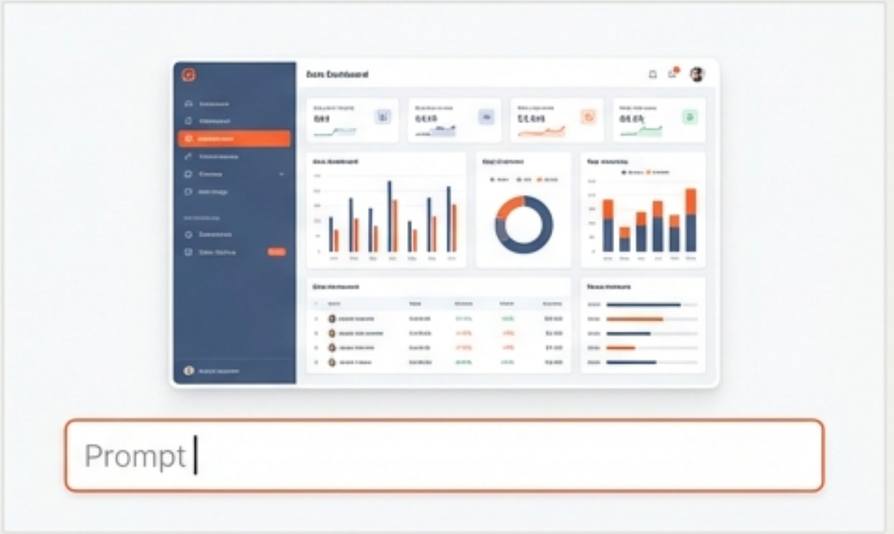
The collapse of the barrier to entry

THE OLD WAY (Weeks)



- Learn Syntax
- Dig through API Docs
- Configure Localhost
- Maintain Frameworks

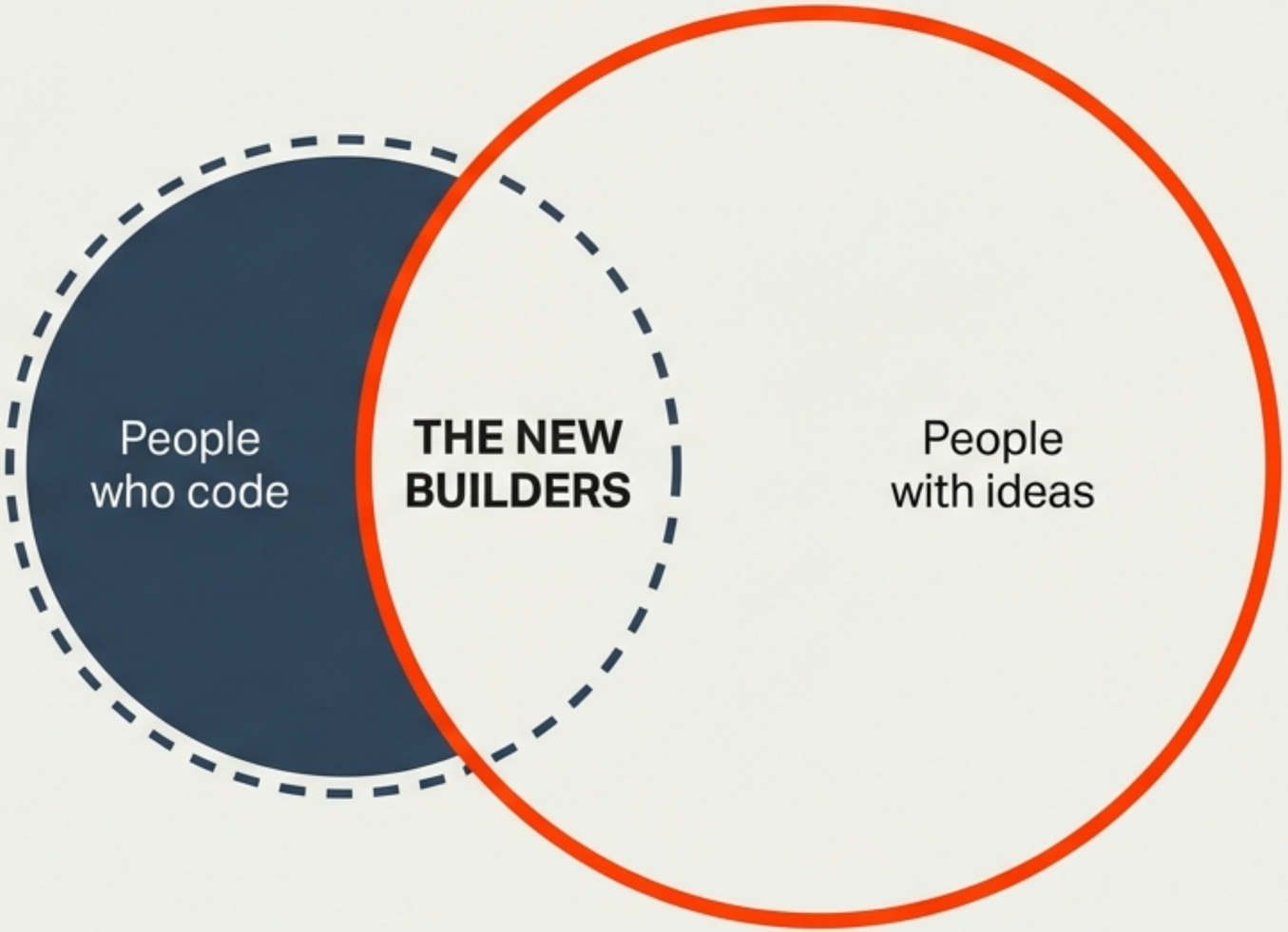
THE NEW WAY (Hours)



- Type a text prompt
- Ship with Cursor / Wabi / Replit
- Zero Friction

Evidence: Tobi's custom MRI dashboard and Marc's movie recommender—tools that previously required commercial teams, now built by individuals.

You no longer need to care about software to build it



Previous State: Builders were a niche subculture—Tech Twitter, Paul Graham readers.

Current State: The addressable market is anyone with a good idea.

Insight: Democratization means the user becomes the maker. Doctors, teachers, and shop owners are finally designing products for problems they actually live.

Software is becoming a medium for expression

Moving beyond Utility. Software is becoming a vehicle for personality, humor, and zeitgeist—just like a tweet or a video.

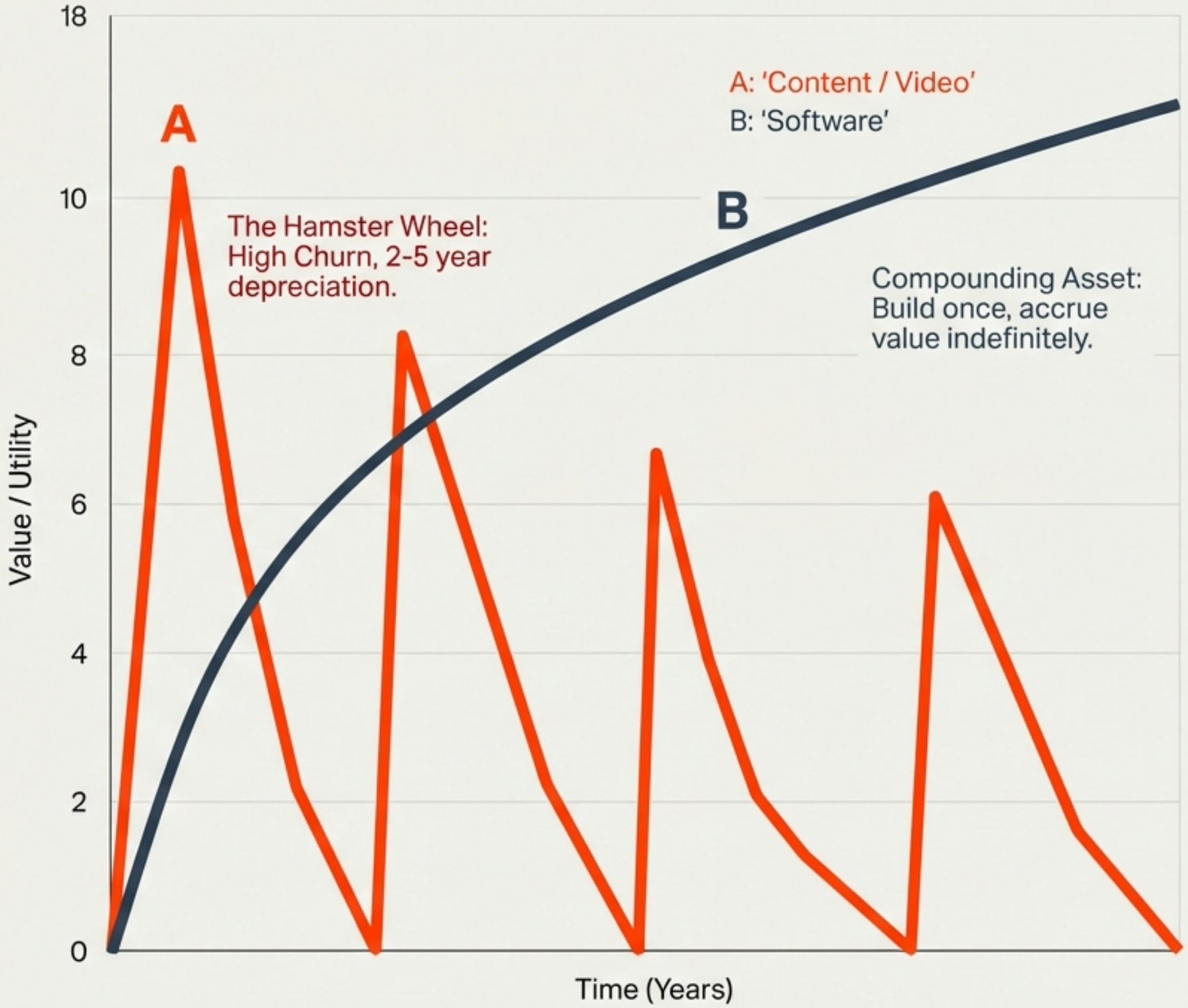
The Shift: People will soon create funny software with about as much effort as drafting a funny post on X. We are witnessing the rise of the "Tiny App".

Riley Walz: Visualizing SF Parking Cops.

Rendering the Epstein Files.

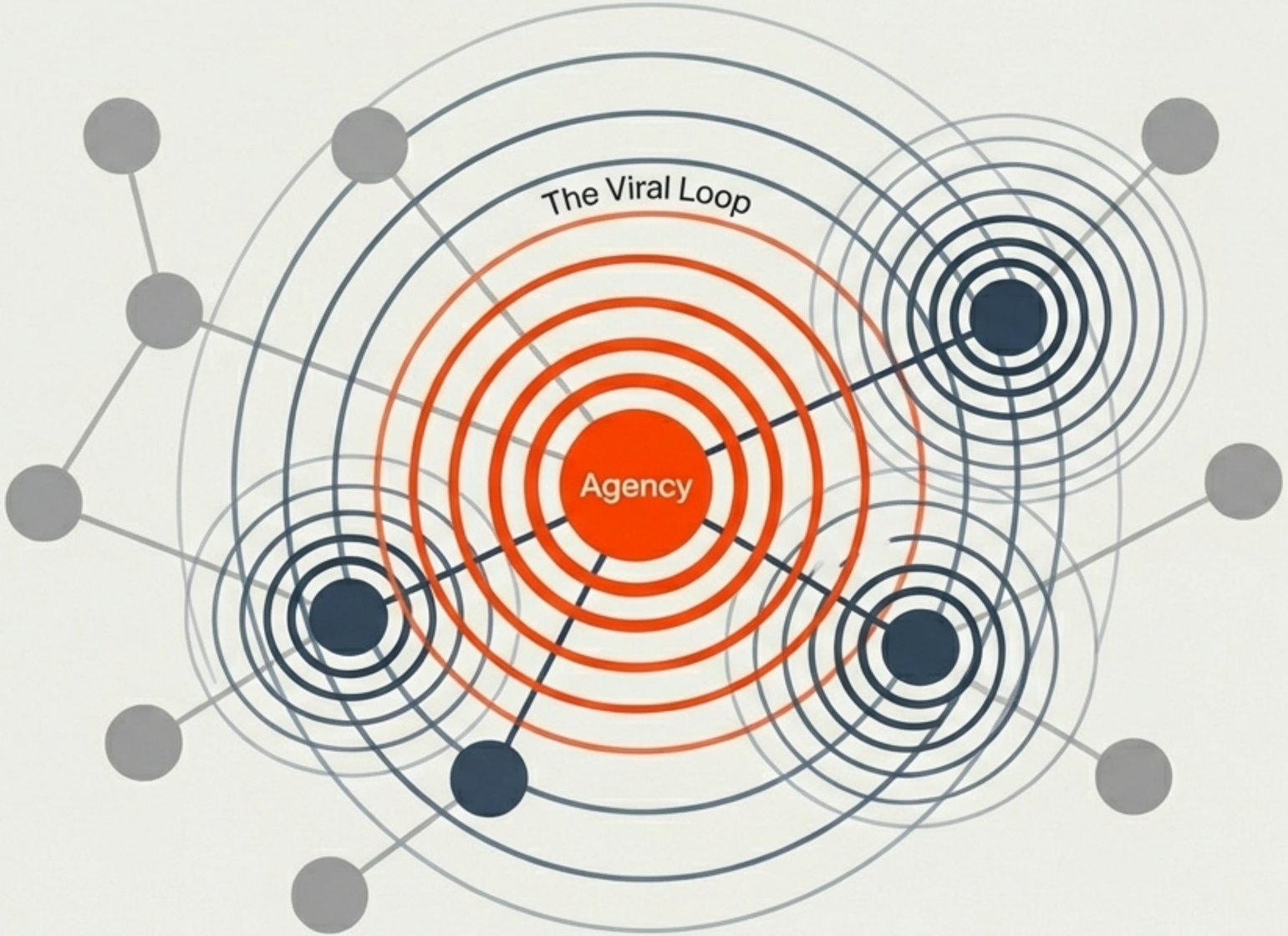
Niche Humor App.

Why Software is a superior asset class to Content



Tiny, expressive software 'posts' can evolve into durable businesses. This is the underappreciated advantage of the medium.

Mimetic Desire & The New ‘Cool’



Why are people building? Because they see their friends doing it.

1. The Viral Loop: Seeing a friend ship an app creates the ‘I can do that too’ impulse.
2. Cultural Shift: It’s no longer about aspiring to be an ‘Influencer’ for fame; it’s about Agency—taking destiny into your own hands.

There is no angst, only envy.

There has never been a better time to be a young person with great ideas.

The Last Mile Problem: Distribution

We have moved from ‘Everyone can build” to ‘Everyone can try,’ but we have not yet reached “Everyone can distribute.’ The ecosystem is waiting for the true “YouTube Platform’—where publishing is as simple as uploading.

The Kids Are Alright

AI has democratized leverage. Tiny, expressive software “posts” will evolve into durable businesses. The barriers are gone.

We are witnessing the birth of a new viral medium where “software” is the message.

The question isn’t ‘can you build it?’ It’s ‘why haven’t you started?’