The End of Engineering as We Know It: Industry Faces Its Biggest Shift

10 min read · Sep 28, 2025

Kapil Viren Ahuja Follow

For leaders and board members, the fundamental redefinition of the engineering role is no longer a debate but an impending reality. Resisting the change is not an option; adapting is the only way forward.

> *Your company's biggest competitive threat isn't a rival — it's your own engineering organization.*

This fundamental shift from large teams to smaller, highly efficient trinities is already driving notable changes. For example, DeepSeek, a small startup with a close-knit team, recently developed an open-source AI model. This model matches the performance of OpenAI's GPT-4—built with a billion-dollar investment—yet it was created with just a $5 million budget.

In my talks with many CTOs and engineering leaders, a pattern emerges. They face resistance to change and higher development costs, even though their teams use Agentic Coding tools. This is the harsh truth they prefer not to hear. Senior engineers with over 15 years of experience are more resistant to the solution than their less experienced colleagues. This isn't just about personal preference; they are too deeply rooted in their approach. They want automation and are not ready to understand the implications.

> *I was on a call today, discussing the two approaches to using Agentic Coding. When I showed them the Phoenix-OS, I got a reaction: "How is the quality?" When I said, "It's meeting all the standards every time," I heard silence.*

This forces a critical question for every board: Is your technology organization an asset driving growth or a legacy liability on your balance sheet? The imperative for change is no longer debated among the world's most influential technology leaders.

For boards and the C-suite, the time for observation is over. The work of transformation must begin now.

> *As Anthropic CEO Dario Amodei puts it,* "It's a race between how fast the technology is getting better and how fast it's integrated into the economy. And I think that's just a volatile and turbulent process."

. . .

This Transformation is here now.

It is an escalating business reality with consequences for valuation, talent, and market position. I am observing three critical impacts that should be on every executive's agenda:

1. **Valuation Risk:** a direct concern for every CEO and CFO. The market is already pricing this transition in. We found that tech-forward companies with demonstrably AI-augmented development processes are trading at a higher revenue multiplier. This valuation gap represents billions in enterprise value at stake for CFOs managing investor relations and CEOs defending market capitalization. Board members I know are wary that their AI story lacks sufficient power.

2. **Talent Crisis:** That should alarm every CHRO and CTO. The war for technical talent is about to be fought on a new front. CHROs face an impossible recruiting mandate, while CTOs watch their best engineers leave for AI-native competitors; a dual crisis that threatens organizational continuity. I am currently driving cultural change in my organization, and it's not easy. This has become our highest priority to stay relevant and understand the nuances of Agentic coding.

3. **Market Speed:** the metric that determines survival for CMOs and Chief Product Officers. In today's economy, speed is a primary driver of market share. There's a prevailing notion that AI drives more coding, yet I have yet to see products releasing features at the promised pace. Our data shows that first-movers who leverage AI-driven development to accelerate their product cycles capture three times more market share than their slower competitors. For CPOs managing product roadmaps, the ability to iterate in days rather than months isn't just operational efficiency — it's the difference between market leadership and obsolescence.

The force driving these outcomes is a fundamental shift that renders resistance futile, transforming even the most complex legacy systems not just a strategic advantage, but a necessity. This article breaks down the nature of this disruption, explains the forces that make it irresistible, and provides a decision framework for leaders to navigate the path forward.

The Catalyst: The $Trillion Disruption

Let me be blunt: if you think AI-assisted development is just another tech trend, you're already behind. The economics have shifted so dramatically that I'm watching companies hemorrhage talent and market share simply because their leaders can't grasp the magnitude of what's happening.

Claude-Code is not a cute experiment anymore. I've watched myself outpace engineers who refuse to learn how these tools work. But here's what really got my attention. 35 days to 3.5 days to write a microservice. Not incremental improvement. A complete reimagining of what's possible. Yes, it took me 4 weeks to set up my Agentic system (Phoenix-OS), but once I trusted it, it worked every time. I rethought everything — Design, testing, deployment—the whole damn pipeline. And I don't think I am done yet.

Your CFO Should Be Losing Sleep Over This

I spend a lot of time with CFOs, and here's what makes them sit up straight: engineering costs are killing profitability. Always have been. We've just accepted that 18–25% of revenue goes to engineering in tech companies. It's like a law of physics.

> *Wrong. Dead wrong.*

Companies that have made this transition? They're running at 8–12% of revenue. Do that math on your own P&L. For a billion-dollar company, that's not a rounding error — $70–130 million annually, straight to the bottom line. And here's the kicker that most analysts miss: every percentage point you shave off engineering costs pumps up your EBITDA by 2.3%. I've seen CFOs literally recalculate this three times because they can't believe it.

> *Why This Isn't Like Cloud or Open Source*

Look, I was there for the cloud migration. Watched companies agonize for years about moving off-premises. Open source? Same story — slow, cautious adoption.

This is different. Viscerally different.

The cloud took a decade to go mainstream. AI coding tools? Three years, tops. Maybe less. The velocity is unprecedented because the value proposition is immediate. You don't need to retrain your infrastructure team. You don't need to rebuild your architecture. You… start coding differently. Tomorrow.

This isn't about whether you believe in AI. The market has already decided. Your only choice is to lead this transformation or become its casualty.

. . .

The New Engineer: A Redefined Role and Value

> *I want to say it's me, but I can't show you.*

I had coffee with a principal engineer last month. Twenty-five years in the industry, built systems that handle billions of transactions. I hoped to hear "*I don't write code anymore, I argue with AI about architecture*". Yet, all I heard was theory, some frameworks, and no knowledge of what's inside. I found a poser (if I may say so) — my bullshit radar was going bonkers.

The engineers who'll thrive are the ones who can think in systems and orchestrate AI agents like a conductor leading a symphony. My role has evolved into something I could not have imagined. I am seeing a new engineer born.

The Architect Emerges

What most people miss: AI amplifies the need for engineering expertise — a different kind of expertise. In the 3rd week of The Grind, I finally see some of my engineers debating patterns and orchestrating Agentic Coding. There are traits that we are building towards:

1. Think in architectures, not algorithms.

2. Debug systems, not syntax;

3. Measure success in business outcomes, not lines of code.

Why Human Expertise Becomes More Valuable, Not Less

> *Coding was never the roadblock. Yes, that's right.*

We have been abstracting frameworks for decades, and writing code is, for the most part, copy-paste. The hardest part is the first two months in any project when we build architecture and patterns. What takes time is ensuring that engineers can replicate consistency all the time. Hence, we devise methods—code reviews, PR reviews, unit testing, etc. This is counterintuitive, but stay with me.

> *When coding becomes commoditized, architecture becomes premium. When syntax is automated, systems thinking becomes scarce.*

AI-generated systems work perfectly… until they don't, until they hit an edge case nobody anticipated. Then, you can't find what was left or what was done under the covers. Polyrepos is another problem in the enterprise.

The Leadership Crisis Nobody's Talking About

Engineering managers are in crisis, and most don't even know it. Their careers have been built on managing tasks, reviewing code, and estimating sprints. I do this in seconds using Phoenix-OS. Suddenly, they're being asked to become strategic thinkers and architectural reviewers. The skill gap is massive. This is why I am trying to hire another engineering manager, which is the wrong role today—very bad.

One VP of Engineering told me, "I spent fifteen years learning to manage engineers. Now I need to learn to manage AI agents AND engineers who manage AI agents. My job completely changed." She's not wrong. And she's one of the ones who gets it.

The future belongs to those who can think bigger, abstract better, and orchestrate rather than implement.

. . .

Navigating The Transition Period

This transformation is going to be messy. Really messy. Not because it's hard, but because people are wooed with marketing videos and CEOs saying they generate 45% of their code using these systems. No one explains the pain behind the final video, how many prompts they wrote, or the back-channel sponsorship to use the underlying tool for free.

> *This is marketing at its finest — selling dreams while hiding nightmares.*

I am looking at 18–24 months of running two parallel realities.

1. Your legacy systems that pay the bills and;
2. Your future systems that will save your company.

> *You're delusional if you think you can flip a switch and be done.*

I'm working with several clients. We are starting incrementally, with only a handful of their engineers using AI tools, and others waiting until we have built an organizational capability with the current set. We are planning for an explosion when we hit the inflection point with the critical mass of engineers who understand how to build the system. We hope that until we hit the inflection point, the other half will not succeed in sabotaging the effort.

You'll run a Frankenstein operation for two years (give or take). New microservices built with AI sit next to legacy systems held together with duct tape and prayers. Your architects will need to design bridges between worlds. The Agentic system needs to understand both worlds, while not taking you bankrupt with the all-new token economy.

I have always called for "transitional architectures," which are ugly by design. Most don't get it. Some dare to ask me what it is, but most bypass the topic. In short, these architectures are not meant to last but to keep you alive while you transform.

Here's where it gets really fun. I've watched companies create these *"AI Centers of Excellence," innovation theaters where executives can point and say, "Look, we're transforming!"* Meanwhile, these labs never integrate with the actual engineering teams. They build cute demos, win internal awards, and sometimes win revenue with clients. But they accomplish exactly nothing.

They are building silos and setting up the whole organization for failure. Only last month, I was talking to a very senior leader (almost 20+ years of experience). They won a new, cutting-edge project, but it was put behind some compliance nonsense when asked to share. While we won some revenue, we lost an excellent opportunity to build AI literacy.

> *This is the recipe for short-term wins and long-term losses*

The real battle? It's being fought elsewhere, on a very different plane. CTOs, Chief Digital Officers, and Heads of Innovation are fighting over who owns the transformation budget. Meanwhile, the CFO wonders why everyone needs $30 million for something that "might work."

Here's my advice: Give it to whoever has P&L responsibility. I run my own P&L, and I am investing in the future. I am willing to share it with everyone in the organization, but I am investing in my survival. This is not an innovation topic that dies in a lab somewhere; this needs to happen on the ground.

. . .

How to Prepare for What's Next

For the Engineers Reading This

Stop protecting your old expertise like it's gold. Full stack is table stakes now. What matters is your ability to think in systems. Deliver the end-to-end solution, understand the process, and understand the product.

Start with Clarity. Being able to dictate in an evident and crisp form to an AI what you want from your architecture, from your system, is an art. Learn how to decompose problems. That is the only way to guide through complex architecture and problem statements. That is the only way to review what AI will generate for you. You have to trust that the AI can code.

Even more important is your mindset shift. Stop thinking like a coder. Please don't come and tell me that you've used AI to automate unit testing. I want to hear what business value you've delivered.

For the Leaders Trying to Navigate This Mess

Stop lying to yourself. Throw out your 3-month PI plan. Start thinking in one-day sprints. What can you transform this quarter to deliver value next quarter? The only investment that matters today is your team's ability to adapt. Plan to give engineers time to experiment. Real-time, not Friday afternoons.

And please, for the love of quarterly earnings, foster a culture where failure is data, not disaster.

Your 12-Month Reality Check

If you do this right, here's what you should see within two years:

1. Feature deployment becomes 4x faster — not because you're coding faster, but because you're eliminating the senior engineer unwilling to transform.
2. Your time from idea to production should collapse from 10 days to 2.

Yesterday, I walked into the office and had an idea of developing a feature in Phoenix. Before I wrapped up my day, I had it ready and used it. In the next five days, it will be part of my release.

> *It took me 3 months of self-conditioning to be here; this is not happening overnight.*

This transformation isn't just about technology or economics. It's about identity. We're asking an entire profession to reimagine itself. Suddenly, "How to Architect" seems so much more relevant. I feel so empowered to discuss this reimagined way of building and architecting systems.

The excellent engineering shift is here. And every day you wait to embrace it is a day your competitors pull further ahead. The choice is yours: evolve or become extinct.

Welcome to the new world. It's messy, uncertain, and thrilling if you're brave enough to embrace it.

Agentic Ai · Generative Ai Use Cases · Thought Leadership · How To Architect

Follow

Published in Generative AI

62K followers · Last published 3 days ago

Stay updated with the latest news, research, and developments in the world of generative AI. We cover everything from AI model updates, comprehensive tutorials, and real-world applications to the broader impact of AI on society. Work with us: jimclydegm@gmail.com

Follow

Written by Kapil Viren Ahuja

91 followers · 36 following

Creating Architectural POVs | Nurturing architects of the future.

No responses yet

Bgerby

What are your thoughts?

More from Kapil Viren Ahuja and Generative AI

In Generative AI by Kapil Viren Ahuja

Top-Down AI Adoption Beats Bottom-Up Every Time (Except When It Doesn’t)

I have been watching leaders struggle with AI adoption for years now, and I need to tell you something that might sound contradictory…

Sep 15 57

In Generative AI by Joe Njenga

15 Ways I’m Using Google’s Nano Banana for UI/UX Design (Like a Pro)

I am not a UI/UX designer, but Google Nano Banana is filling the gap, and that doesn't mean I'm becoming one overnight.

Oct 9 514 7

In Generative AI by Dr. GenAI

Hierarchical Reasoning Model (HRM): a tiny brain that embarrasses giant LLMs

HRM is a 27M-parameter model trained in hours that beats giant LLMs on reasoning puzzles, proving architecture can trump scale.

Sep 7 306 6

In Towards AI by Kapil Viren Ahuja

CTO’s ADAPT System: My Five Bets for the Agentic Engineering Era

A Principal’s Bets for Thriving in the Agentic Engineering Era

Oct 5 68

See all from Kapil Viren Ahuja

See all from Generative AI

Recommended from Medium

In The Generator by Thomas Smith

OpenAI Finally Admits the Real Reason it Crippled GPT-5

And what it’s doing to make things right

Oct 19 939 33

In AI Advances by Michael Dain

AI uncovered the secret of everything

And I'm not convinced that its wrong… yet.

5d ago 153 1

In Generative AI by Avijnan Chatterjee

This Free AI Tool Does What ChatGPT Can't

Discover Google AI Studio's hidden features: system prompts, Compare Mode, and real-time screen sharing. Free tutorial for analyzing…

6d ago 121

In AI Software Engineer by Joe Njenga

OpenAI Launches a Chrome Killer (ChatGPT Atlas) But Everyone is Skeptical!

ChatGPT just launched a web browser and it’s not what you think.

4d ago 63 2

In Towards Deep Learning by Sumit Pandey

Why Everyone Will Want DGX Spark on Their Desk — Yes, Everyone

I just saw this picture today and was amazed, I've been waiting for this moment for a long time. (No, it's not Elon.) It's that tiny...

Oct 14 53 11

In Analyst's corner by Mahathidhulipala

Why Everyone Got AI Wrong (And What Actually Matters)

What I learned about artificial intelligence by watching it fail to do what people expected, and succeed at things nobody predicted

Oct 14 441 12

See more recommendations