The State of Vibe Coding

How **English** became the world's fastest-growing programming language

01

How vibe coding became the new way of working

Intro

Earlier this year, vibe coding was just emerging. Andrej Karpathy, who co-founded OpenAI, coined the term in February 2025, describing it as an AI-led coding experience “where you fully give in to the vibes, embrace exponentials, and forget that the code even exists.”

In March of 2025, Garry Tan said AI-powered teams of under 10 people now do what once took 100 engineers. Three months later, Sundar Pichai was spending his spare time building webpages with bare-minimum coding. And by late summer, Larry Page was building with v0.

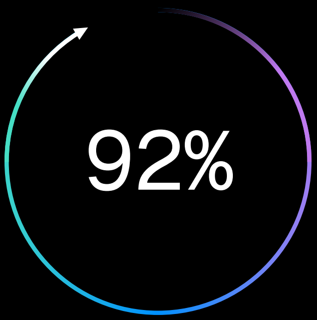
Why?

Because English is becoming the world’s most powerful programming language—and we’re all witnessing it from the front row.

> “I’m building a project or web app, but it’s not really coding—I just see stuff, say stuff, run stuff, and [copy-paste] stuff, and it mostly works.”

—Andrej Karpathy, Co-Founder of OpenAI (X)

Over the past six months, vibe coding has been rapidly revolutionizing the industry, reinventing the “rules” about who can code and how fast they can do it.



of U.S. developers use AI coding tools every day

Vibe coding tools are hitting

$100M+

annual recurring revenue (ARR)

in just months

25% of Y Combinator startups

rely heavily on AI-generated code for their core systems

90% of code will be AI-generated

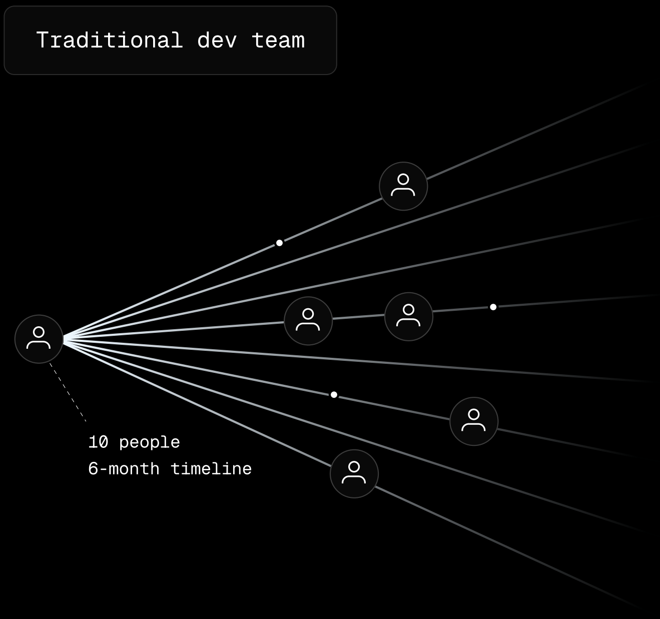
in as little as three months, according to Anthropic CEO Dario Amodei

It’s not just hype—it’s happening everywhere

Vibe coding’s impact goes way beyond better productivity or nine-figure ARR.

It’s completely reinventing how we work, shifting the fundamental unit of business innovation from “team” to “individual.” This shift affects both the economic and cultural aspects of internet businesses.

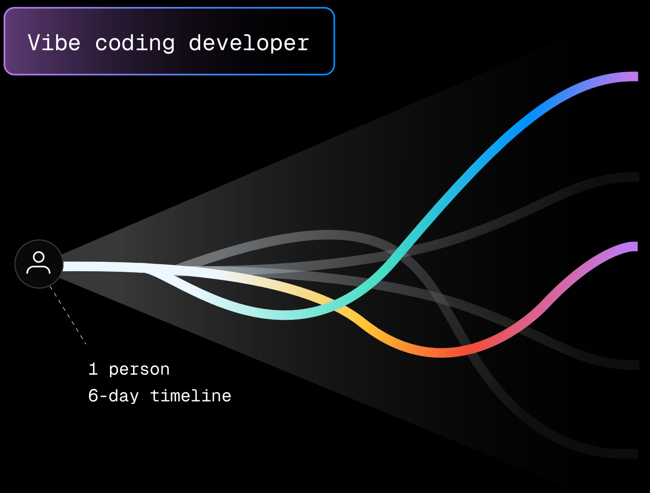
And it’s not just “coming soon”—it’s already here.



The scale of vibe coding is pretty wild—but there are caveats

People new to development—recent grads, product managers, designers—suddenly have supercharged coding capabilities. But they’re coding without the traditional guardrails that experienced engineers rely on. There are serious privacy and security considerations that often get overlooked when you can create working code in a matter of minutes. When you build fast, critical security audits, data handling protocols, and access controls can fall by the wayside.

This represents an urgent, industry-wide need to build and maintain the appropriate guardrails for vibe coding.



02

From zero to revolution in six months

WHAT IT IS

There's nothing fast about cultural evolution.

It's taken humans millions of years of adaptation and iteration to get where we are now. But vibe coding's cultural revolution? It's running on fast-forward.

The evolution timeline

Early 2010s

Bubble, Airtable, and Retool launch as the world's first no-code tools.

2017

Machine learning and AI models push the revolution forward, deeming 2017 "The Year of AI."

2021

GitHub Copilot launches as the industry's first AI pair programmer.

2024

Cursor's Composer marks the true emergence of vibe coding.

2025

Startups like Lovable, Bolt, Replit, and **v0 make vibe coding accessible to everyone.** WYSIWYG 2.0 emerges—what you **say** is what you get, instantly.

How vibe coding works: WYSIWYG for any idea

Unlike traditional AI-assisted coding or pair programming, vibe coding means you:

- Describe what you want in plain English
- Let AI suggestions guide projects with little (or no) code review
- Focus on iteration and experimentation
- Trust the AI to handle heavy lifts like syntax, structure, and implementation

To vibe code, the only language you need is the one you use every day. **This shift transforms coding from a technical skill requiring years of training into a conversational interface that anyone can master in hours.**

03

You don’t need to be an engineer to vibe code

WHO’S DOING IT

Vibe coding is flipping coding culture on its head, letting people who don’t “speak” programming languages tap into all the good stuff.

Vibe coding’s new power users are:

- **Entrepreneurs** with big ideas but no technical background
- **Marketing teams** that need customer databases built in minutes instead of days
- **Product managers** who want to prototype actual applications instead of just wireframes
- **Designers** who want to explore beyond a flat view of their product

> “[I]t’s now easy to spin up custom software with almost no effort. … Software creation used to be constrained by ROI. Now it’s constrained only by imagination, and that’s a much more interesting limit.”

—Anish Acharya, General Partner at Andreessen Horowitz (a16z) (X)

A vibe-based workplace

Individual users aren’t the only ones experiencing the vibe coding shift. The biggest platforms in the world have been working overtime to catch up—both using AI for their own development and revamping their market offerings to include vibe coding solutions.

Google

30%+

of new code at Google is generated by AI (before being reviewed and accepted by company engineers)

Microsoft

Similar figures were shared by Microsoft CEO Satya Nadella at a recent conference

amazon

Amazon launched Kiro as its vibe coding solution

Apple

Apple’s developing a new version of Xcode for internal use in partnership with Anthropic

Vanguard

Vanguard is experimenting with vibe coding to prototype new webpages, speeding up design by 40%

This shift to natural language programming isn't just changing how we code. It's changing what gets built, who can build it, and the future of digital creation as we know it.

Ideas will become programming's new currency, with projects limited only by what users can imagine and articulate.

Those who wait for vibe coding to "mature" are going to be displaced by those who don't.

Three key benefits everyone's seeing

While the big players are doing the most visible vibe coding adoption, teams of all sizes are seeing three benefits:

Faster iteration cycles

Vibe coding allows prototypes to be created in a matter of hours, so product teams can do more without adding headcount. Using v0 can slash iteration cycles by up to 50%.

Individual productivity scaling

Using v0 bridges the gap between design and code. PMs can design, designers can code, and sales reps can build products.

Cost savings

Vibe coding requires fewer specialized developers, less maintenance overhead, resulting in fewer bugs and lighter roadmaps.

> "95% [of code lines are] going to be AI-generated. Very little is going to be [line by line] human-written code. Now, that doesn't mean that the AI is doing the software engineering job, and so I think the more **important and interesting part of authorship** is still going to be entirely human."

—Kevin Scott, CTO of Microsoft (Business Insider)

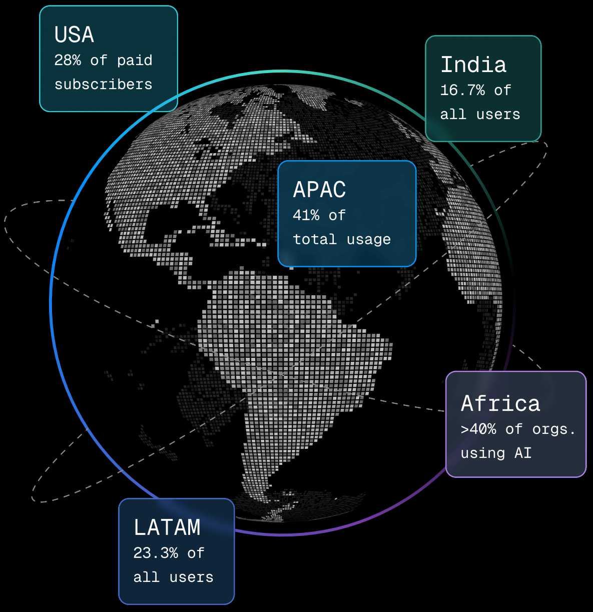
Global adoption patterns

Vibe coding is spreading globally, but adoption looks different everywhere you look.

The following insights are based on internal v0 usage data and user analytics, unless otherwise noted.

Trends in vibe coding across the world

Here are the regions leading the charge in vibe coding—and what's driving them.



USA

While the U.S. ranks second in overall usage, American users are the most willing to pay for AI coding tools. American companies are buying this technology at high rates, and many Fortune 2000 companies are actively paying for these tools.

India

India's massive developer community has embraced AI coding tools, benefiting from strong English proficiency and an entrepreneurial tech culture. India leads in overall vibe coding adoption, accounting for 16.7% of all users.

Europe

European adoption is strong but measured, with some EU companies avoiding external AI services for code generation to ensure compliance with the region's evolving AI regulations.

Asia Pacific

APAC represents an impressive 41% of total usage, driven primarily by India, Japan, Pakistan, and Indonesia. Notably, China has minimal presence, which is typical of a market focused on developing domestic AI alternatives like Huawei's PanGu and Baidu's ERNIE.

Africa

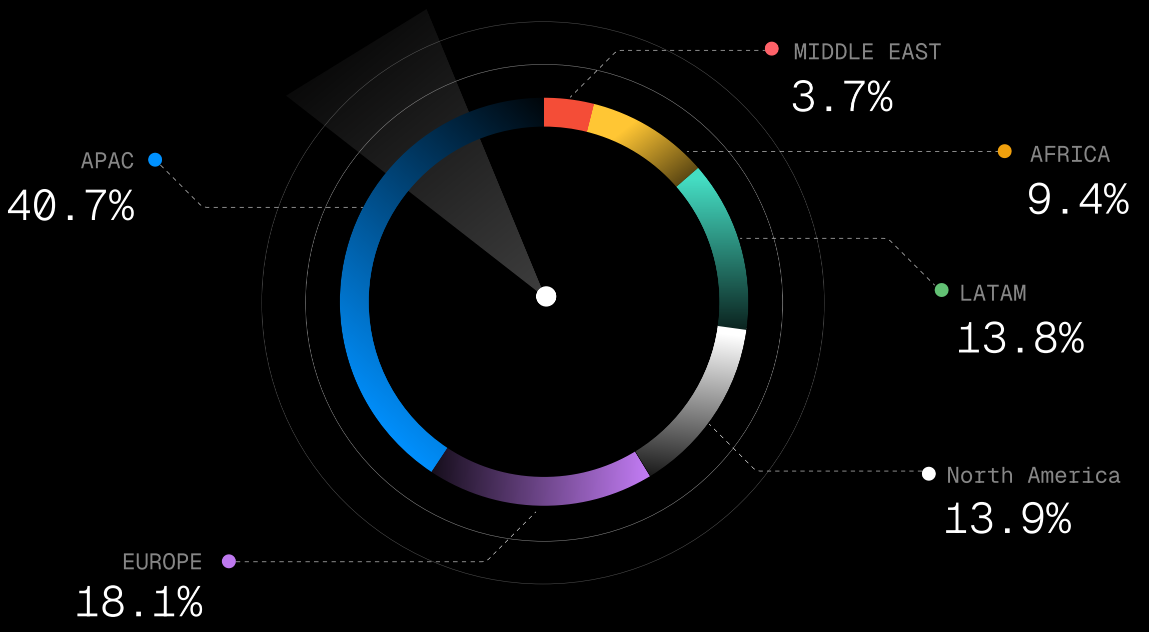
More than 40% of organizations across Africa are either experimenting with or have implemented significant AI solutions, positioning the continent as a global leader in global AI adoption for translation, academics, and other applications.

Latin America

While Latin America ranks fourth globally, the region is making a coordinated push with 12+ countries collaborating on Latam-GPT—a regional language model designed to capture the "region's diverse cultures and linguistic nuances."

Global distribution of v0 users

Unique users by region in the past 30 days



Data as of August 18, 2025

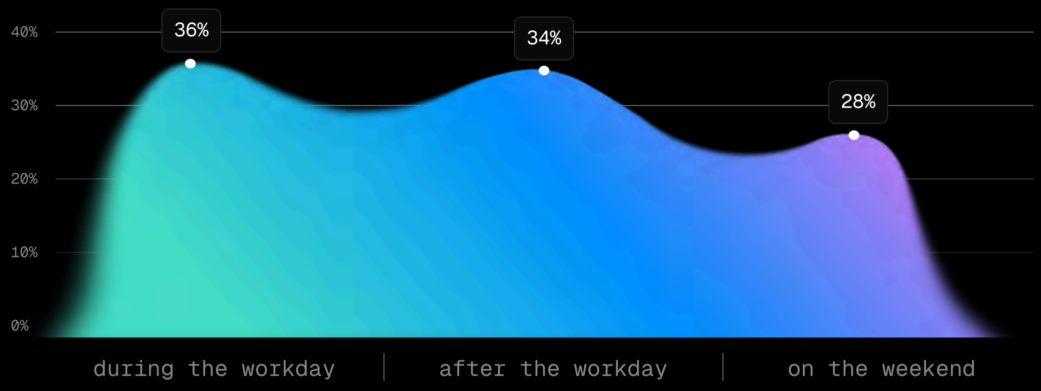
Trends in individual vibe coding

Who's using these tools:

63%
non-developers

37%
developers

When they vibe code:



Productivity appears to be correlated with uninterrupted focus time—users operating between 9 am–5 pm (local time) complete more sessions compared to moonlighters and weekend warriors.

What they're vibe coding:

44%
UI generation (e.g., build a form; design a component or layout)

20%
sophisticated applications (e.g., build an ecommerce site)

11%
personal websites & portfolios

How they prompt:

Developers tend to mention specific frameworks in the prompts:

+ "Do these for me using typescript and react-chart or apex chart"

+ "In vue there is v-bind directive, what would be the similar to do it in next.js/react?"

+ "create a sophisticated file conversion website with SEO optimization. this website will be in blogger. use only html css and js no react."

Non-developers use more conversational, goal-focused prompts:

+ "hi i need you to extract the articles, and the keyword and create a table in a csv with these, can you do that?"

+ "I need a reservation system for a restaurant"

+ "Can you create an app for identifying statistics, trends, and providing performance forecasts for the top MLB players at every position."

04

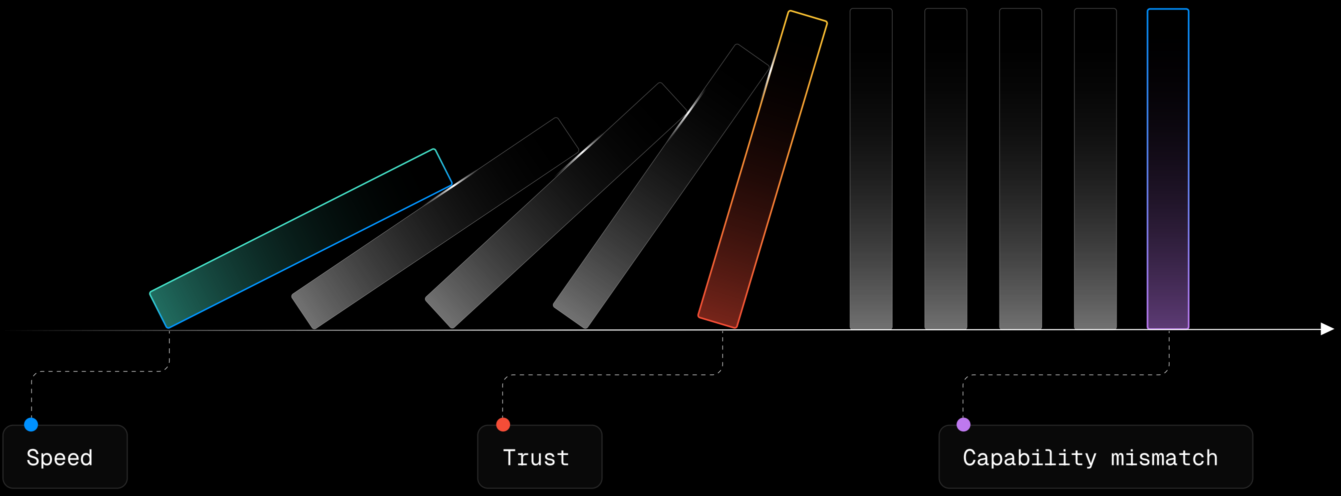
Rapid expansion has created a problem

WHAT COULD GO WRONG

As English becomes the world's newest coding language, millions of people gain the power to start coding without any experience. While that opens up incredible opportunities, it also creates some pretty serious risks.

Vibe coding's speed cuts out traditional buffer time for security review, code audits, and vulnerability testing that the industry relies on to stay safe. And most people don't have the security training that this new responsibility requires.

The problem with expansion



Speed

- New capability → Project development cycles are compressed from weeks (or months) to minutes
- New risk → Increased coding speed eliminates traditional security checkpoints

Trust

- New capability → Vibe coding means you don't really think about the code—AI wrote it, so it should work
- New risk → AI writes code that works, not code that's guaranteed to be secure

Capability mismatch

- New capability → Anybody can vibe code, regardless of coding experience
- New risk → Users with no experience can jump into projects without safety expertise

> "If you are vibing, you're not looking at the code.... That is very different from using AI to write code that is reviewed, tested, and maintained over time."

—Lee Robinson, VP of Developer Experience at Cursor AI (X)

The big three vulnerability patterns

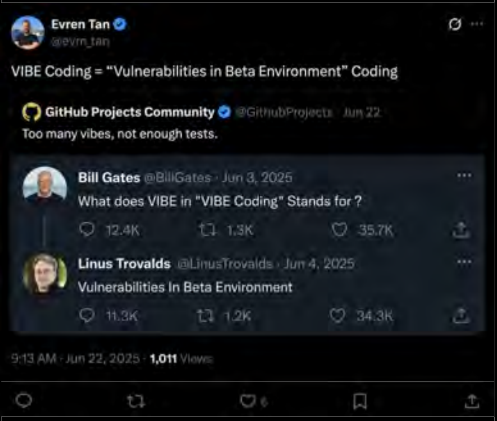
From websites and apps to online storefronts and custom platforms, the potential of vibe coding is growing every day.

But despite its incredible capabilities, vibe coding keeps hitting the same three vulnerabilities multiple times over.

Vulnerability #1

Exposing secrets

AI defaults to writing client-side code unless explicitly instructed not to. If AI writes code that stores confidential information—like API keys, or passwords—on the client instead of the server, those secrets are accessible to anyone inspecting the webpage source code.



Evren Tan @evrn_tan

VIBE Coding = "Vulnerabilities in Beta Environment" Coding

GitHub Projects Community @GithubProjects · Jun 22

Too many vibes, not enough tests.

Bill Gates @BillGates · Jun 3, 2025

What does VIBE in "VIBE Coding" Stands for ?

12.4K 1.3K 35.7K

Linus Trovalds @LinusTrovalds · Jun 4, 2025

Vulnerabilities In Beta Environment

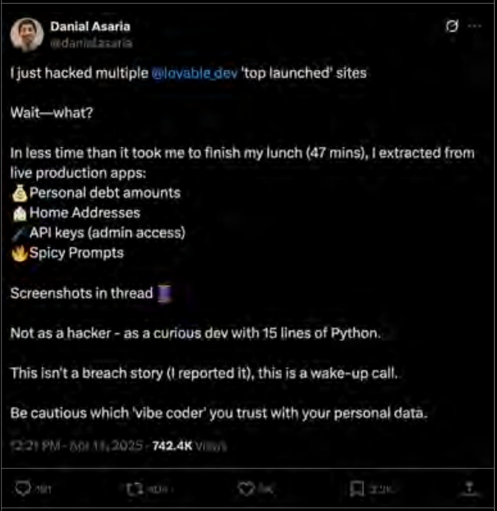
11.3K 1.2K 34.3K

9:13 AM · Jun 22, 2025 · 1,011 Views

Vulnerability #2

Access misconfigurations

AI-generated code can also contain severe access misconfigurations, such as storing sensitive data at publicly accessible URLs, creating databases that can be written to by the public, and writing API routes without authentication. This can expose confidential data to unauthorized users.



Danial Asaria @danialasaria

I just hacked multiple @lovable_dev 'top launched' sites

Wait—what?

In less time than it took me to finish my lunch (47 mins), I extracted from live production apps:

💰Personal debt amounts

🏠Home Addresses

🔑API keys (admin access)

🔥Spicy Prompts

Screenshots in thread 🧵

Not as a hacker - as a curious dev with 15 lines of Python.

This isn't a breach story (I reported it), this is a wake-up call.

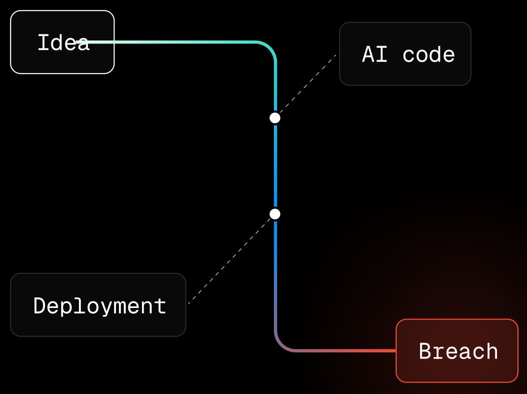
Be cautious which 'vibe coder' you trust with your personal data.

12:21 PM · Apr 11, 2025 · 742.4K Views

Vulnerability #3

Hardcoded credentials

AI can hardcode privileged credentials for testing or debugging purposes into an app—sometimes without a user's knowledge. This creates a dangerous attack vector for malicious actors who can find and use these hardcoded credentials to access sensitive information that would otherwise be protected behind auth.



> "A vibe-coding PM should focus on learning which auth providers to trust and implement rather than building custom solutions."

—Claire Vo, Founder of ChatPRD (v0 AMA)

Building the guardrails

“It’s a failure of the product to not automatically prevent security issues.”

—Zeb Hermann, General Manager of v0

No matter how important security is to vibe coding, the reality is this:

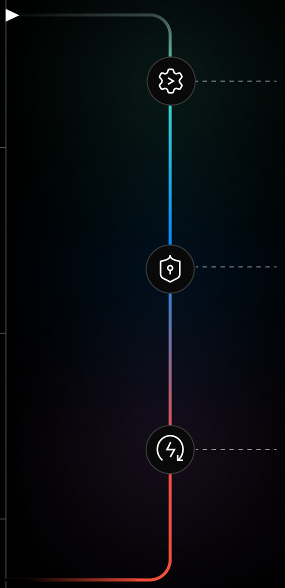
It would simply take too long to train every vibe coder on every security risk that exists.

So, if it’s not the user’s responsibility to mitigate security risks, whose is it?

The product’s.

▲ The product must handle what users can’t be expected to master.

Here are three security green flags that businesses should look out for when exploring vibe coding on a new platform:



Automated intervention systems

Look for a platform that implements real-time scanning that automatically detects common vulnerability patterns in AI-generated code and blocks insecure deployments before they can go live

Security-by-default architecture

Pick a platform that commits to “making the secure path the easy path” through automatic HTTPS, secure headers, sandboxed environments, and default configurations that prioritize code safety over convenience

Control mechanism innovation

Unlike traditional document sharing, where files can be distributed without oversight, look for modern development platforms with kill-switch capabilities that can instantly disable projects when security issues are discovered

The most successful vibe coding platforms will be the ones building security education directly into their development interfaces, providing users with real-time guidance to help them understand (and avoid) common vulnerability patterns.

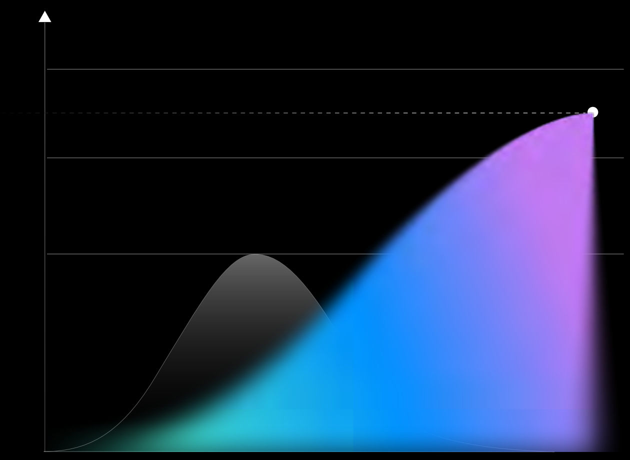
> “…AI-assisted developers produced three to four times more code than their unassisted peers, but also generated ten times more security issues.”

Source: X, on security findings from research conducted by Apiiro, an agentic application security platform

<If we can solve the security challenge, what comes next is remarkable.>

05

The floodgates are open

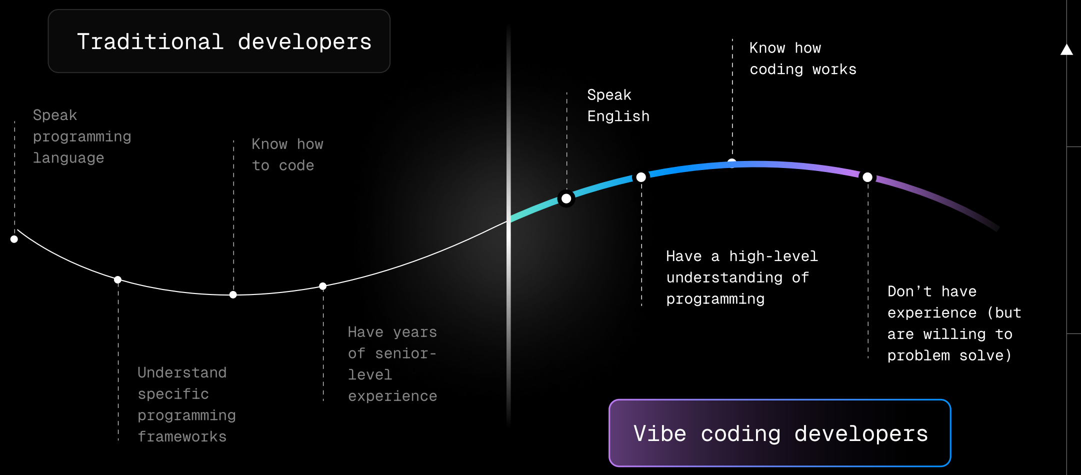


AI-generated code enables virtually any idea to come to life with limited coding experience. With ideas being vibe coding's hottest commodity, traditional organizational structures are going to shift towards people who are creative thinkers—not necessarily those with years of experience.

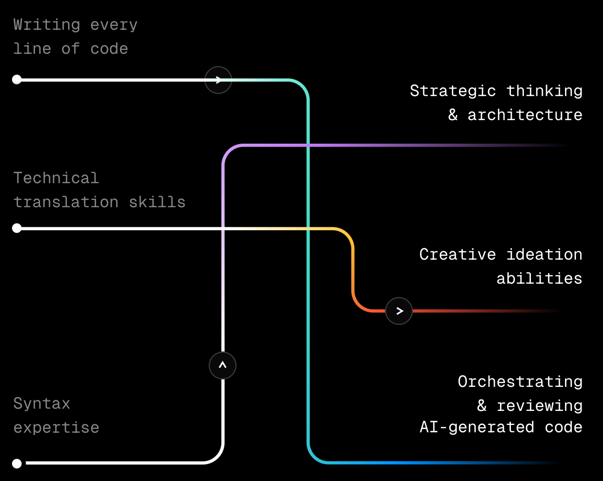
Programming workplaces aren't going to look the same anymore. They're getting leaner, more agile, and less "experienced developer" heavy—a trend called "The Great Flattening." Think of it as what happens when companies ditch the middle management bloat and let the people actually doing the work talk directly to the decision-makers. It's been happening across industries for years, and now it's coming for dev teams.

The developers who stick around won't have the same job descriptions, either. Problem-solving abilities and output are being favored over technical translation skills, and "knowing how to be creative" is racing ahead of "knowing how to code."

Within the next decade, we expect to see developer roles emphasizing higher-level skills, with programmatic understanding becoming less important than ideation abilities. And, with one AI-enabled orchestrator now able to outperform teams with ten times as many developers, things will only get flatter from here.


Traditional developers
Speak programming language
Know how to code
Understand specific programming frameworks
Have years of senior-level experience
Speak English
Know how coding works
Have a high-level understanding of programming
Don't have experience (but are willing to problem solve)
Vibe coding developers

Developer roles are evolving from:


Writing every line of code
Technical translation skills
Syntax expertise
Strategic thinking & architecture
Creative ideation abilities
Orchestrating & reviewing AI-generated code

> "You have to think about [vibe coding] as a collaboration with infrastructure and platforms that already exist, most of which are human-made."

— Guillermo Rauch, CEO of Vercel (Best of the Pod)

> "One of the biggest barriers to [software development] is the incredibly steep initial learning curve—vibe coding shaves that initial barrier down to almost flat."

— Simon Willison, Founder of Datasette (Simon Willison's Weblog)

Five future predictions

Vibe coding is here to stay. But that doesn't mean that the vibe coding of today will be the vibe coding of tomorrow. Here are five ways we think vibe coding will be impacted and shaped in the future:

01 Recipes & Lego blocks

Microsoft's Sam Schillace has criticized pure vibe coding, calling AI-generated code "sloppy, long, and fragile." He advocates for a "Lego block" approach—coding in small, manageable "recipes" that AI handles more reliably than complex requests. This works especially well for single-use apps or features too complex to integrate into larger projects.

02 More external resistance

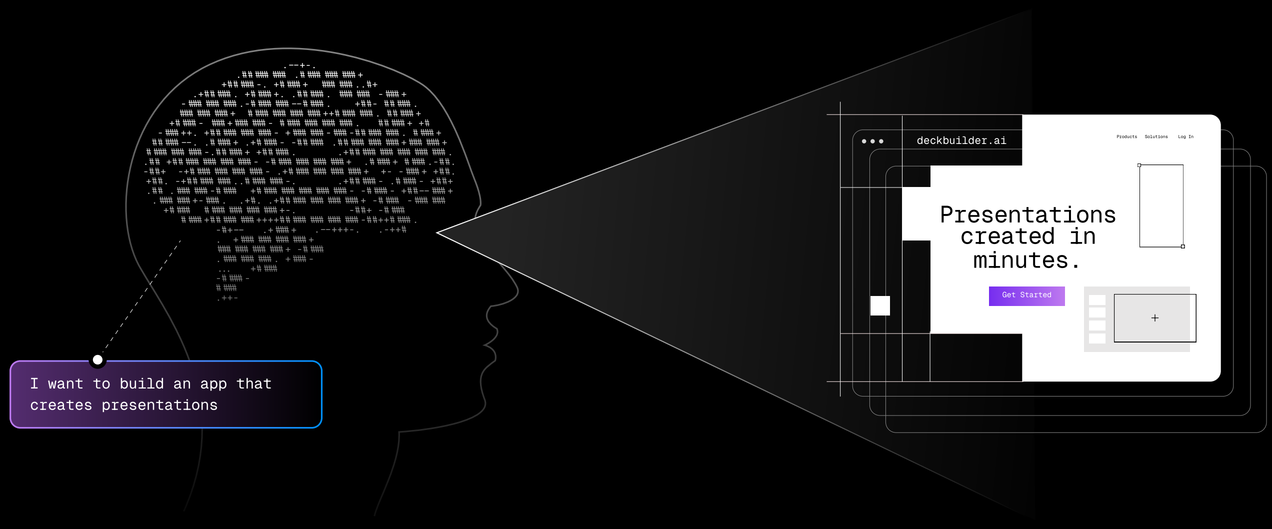
Internal best practices, mandates, regulations, and industry standards will constrain adoption. The EU is already revamping its AI regulations, and we expect more limitations around intellectual property and government oversight.

03 Growing communities

As vibe coders multiply, we expect sharing platforms and social communities to emerge globally, connecting practitioners and sharing best practices. This collaboration could even see a boom in the number of startups born over the coming years.

04 Less competition

Since AI can generate standard solutions, companies may open-source AI-generated components rather than keeping basic code proprietary. This creates a cycle: more open code becomes training data, making AI even better.

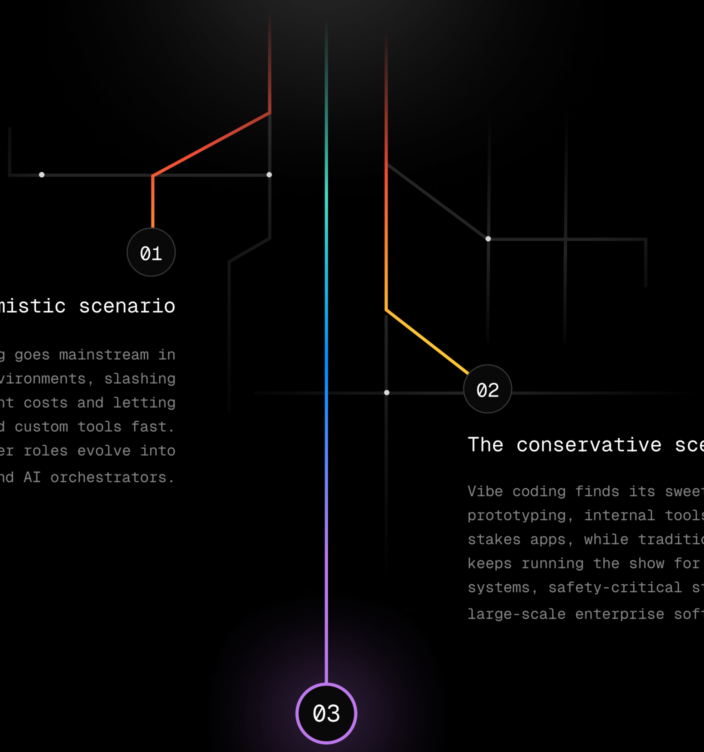

I want to build an app that creates presentations
deckbuilder.ai
Products Solutions Log In
Presentations created in minutes.
Get Started
+

05 Communication replaces knowing the manual

Vibe coding shifts the focus from knowing how to write code to knowing how to describe what you want to build. In this world, traditional product documentation and coding workflows are likely to be replaced by direct collaboration with AI—the key ingredient is having a crisp vision of what you're trying to create, and being able to communicate that to the AI in a way it can understand.

Three future scenarios

The way we see it, there are three evolutionary paths for the future of vibe coding.



01

The optimistic scenario

Vibe coding goes mainstream in enterprise environments, slashing software development costs and letting every company build custom tools fast. Developer roles evolve into system architects and AI orchestrators.

02

The conservative scenario

Vibe coding finds its sweet spot in prototyping, internal tools, and low-stakes apps, while traditional coding keeps running the show for complex systems, safety-critical stuff, and large-scale enterprise software.

03

The likely middle path

By the late 2020s, vibe coding will simply be how development tools work—rather than a separate feature to consider. AI-assisted (or AI-generated) code will become as routine as syntax highlighting, used universally without fanfare.

06

Your move

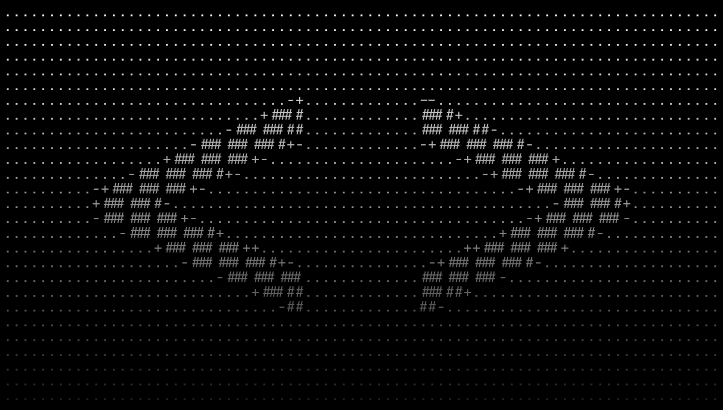
WHAT YOU SHOULD DO

The vibe coding transformation is happening right here, right now.

It’s no longer a question of whether vibe coding will reshape business—it’s whether your organization will build the right guardrails to harness it safely and effectively.

v0 by Vercel is right there with you in the vibe coding revolution, providing the developer tools and cloud infrastructure to build, scale, and secure a faster, more personalized web.

With single-tap deployment, global connectivity in seconds, and design features that enable everyone to code, you can start shipping the best products instantly. Plus, our security features mean you’re contributing to building the proper guardrails with every project you do.



THE CHOICE IS SIMPLE >

Embrace the vibe coding revolution with proper guardrails or watch others build the future while you’re still learning syntax.

v0

Transform prompts into working applications and user interfaces.

Try v0.app