

<https://www.forbes.com/sites/deloitte/2025/04/28/identify-signal-from-the-noise-ai-agents-and-enterprise-management/>

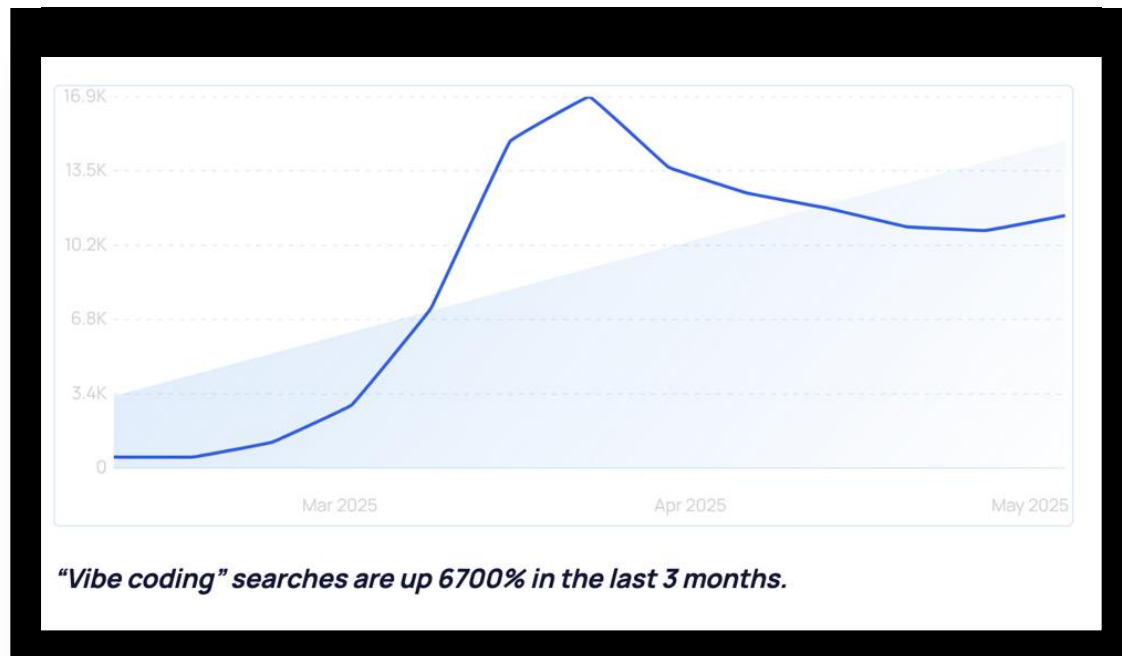
Y Combinator's AI Revolution And The Rise Startups Built By Vibe Coding

By [Josipa Majic Predin](#),

I'm a founder, writer and lecturer focusing on VC funds.

[Follow Author](#)

May 18, 2025, 04.33am EDT



Vibe Coding searches, source: <https://explodingtopics.com/blog/vibe-coding>

"Vibe coding." The term might sound like the latest startup slang or a Gen Z meme, but behind the catchy phrase lies a rapidly evolving trend in how software is written—and who's writing it. At its core, vibe coding refers to using large language models (LLMs) like ChatGPT to generate code from natural language prompts, effectively translating intentions into functional software with minimal traditional programming.

Interest in the term [has exploded](#)—searches are up more than 6700% in just three months. But hype alone doesn't write software—or maintain it. As developers, startups, and enterprises rush to explore the benefits of AI-assisted programming, the question remains: Is vibe coding the future of software development—or just a compelling tool with hard limits?

The Promise: Startups Scale Fast with AI at the Helm

In Silicon Valley's high-stakes ecosystem, time is everything. That might be why [vibe coding](#) is finding early traction among [Y Combinator](#) startups. According to [Garry Tan](#), CEO of the famed accelerator, roughly 25% of companies in the most recent batch are using AI to generate 95% or more of their code.

That's no small experiment. Y Combinator alumni include Stripe, Airbnb, and DoorDash—companies that grew into giants on the strength of well-written code. This latest cohort, heavily tilted toward AI-based ventures (about 80% of them), is betting that LLMs can now handle much of that lift.

Tan cites staggering growth figures, with some startups growing 10% week over week—not just the top one or two, but across the board. That momentum is largely credited to faster development cycles enabled by AI coding tools. “It's like every founder just hired a superhuman engineer,” one YC mentor quipped.

The Reality Check: Benchmarks Tell a Mixed Story

But just how good is AI-generated code?

Enter benchmarks. Tools like SWE-Bench and SWE-PolyBench test AI models on hundreds of programming tasks and bug-fixing scenarios. In 2023, LLMs passed about 5% of SWE-Bench's challenges. Today, the best score over 60%. That's impressive—but still well short of expert human performance.

And results vary. On Amazon's SWE-PolyBench, top models only solve 22.6% of problems. Meanwhile, on Artificial Analysis's Coding Index, the best model scores a 63—compared to 96 on its Math Index, suggesting AI is still better at formulas than functional code.

Even benchmarks can be misleading, since outcomes are highly sensitive to the types of tasks included. “We have a moving target problem,” noted one researcher. “Each benchmark measures a different idea of ‘intelligence’—and they all evolve.”

Coding for the Rest of Us?

Still, the democratizing potential of vibe coding is hard to ignore. Non-programmers are now experimenting with tools like ChatGPT to build functioning apps and games.

One amateur coder described asking ChatGPT to write a basic space exploration simulation. The AI-generated Python code worked on the first try. When asked for upgrades—like graphics or story branches—ChatGPT complied, building progressively more complex iterations.

If that sounds like a toy example, that's partly the point. It shows what AI can do for beginners: remove friction, provide instant feedback, and enable creation without formal training. This could unlock massive latent demand for software—from artists to entrepreneurs—who simply couldn't code before.

The Limits: Debugging, Complexity, and Code Volume

Of course, building toy apps is one thing. Writing production-ready, enterprise-scale code is quite another.

Debugging remains a critical bottleneck. Interest in "AI debugging" has risen 248% in the last two years, and for good reason: when AI-generated code breaks, the solution isn't always obvious—even to the AI itself.

Microsoft is tackling this with [Debug-Gym](#), a training system to help LLMs learn how to fix code like human developers do, using multi-step reasoning rather than just pattern matching. Early tests show notable improvements on benchmarks like SWE-bench Lite, but experts caution that robust debugging still requires human oversight.

And there's a growing volume problem. The easier it becomes to generate code, the more code gets created—often without careful documentation or review. This increases the need for tools like [Lightrun](#), which offers real-time code observability for both human- and AI-written software. Backed by Microsoft and Salesforce, Lightrun claims a 763% ROI over three years and promises up to 20% productivity gains.

Still, even tools like these presume a developer is there to interpret and act on the data.

AI Agents: From Co-Pilot to Captain?

Some startups are pushing toward a bigger leap: AI agents capable of completing full tasks autonomously. Interest in “AI agents” has surged 1113% in two years, fueled by platforms like *Manus*, which aims to bridge “the gap between conception and execution.”

But can AI truly replace a human assistant or engineer?

Not yet. *Manus*, while impressive in planning tasks like creating itineraries or course outlines, failed when asked to execute real-world actions—like booking flights or making reservations. ChatGPT performed comparably in ideation, suggesting the real bottleneck is execution, not creativity.

In software development, the same applies. AI can suggest functions, explain logic, even scaffold entire apps. But without a human in the loop, the risk of bugs, hallucinations, and poor architecture increases exponentially.

The Toolchain: Who’s Winning the Vibe Coding Race?

Developers interested in exploring AI-assisted coding now have a buffet of tools to choose from.

ChatGPT’s o4-Mini High model currently tops many coding benchmarks, with OpenAI’s suite dominating LiveBench’s leaderboard. Close contenders include **DeepSeek R1**, **Claude**, and **Grok**.

Third-party platforms are also innovating. *Cursor*, a sleek AI-native IDE, lets users choose from multiple LLMs and integrates AI directly into the development workflow. The company hit \$200 million in annual recurring revenue (ARR) just months after launch.

Meanwhile, *GitHub Copilot Workspace* now enables plain-language prompts across the entire software development process, from ideation to deployment. GitHub’s ambition? One billion developers.

A New Era—With a Human Core

The rise of vibe coding isn't just a tech trend; it's a shift in how we think about programming. AI is no longer a backseat driver—it's increasingly in the passenger seat, offering directions, fixing the map, and suggesting shortcuts.

But the driver's seat still belongs to humans. At least for now.

While AI can accelerate and democratize software creation, experts caution against full automation. Understanding what the code is doing—and what it *should* be doing—is more essential than ever. Without that clarity, we risk creating software systems so opaque even their creators can't fix them.

In short: vibe coding is powerful, but not infallible. And as with most revolutions, its real impact will depend not just on what the tools can do—but how wisely we choose to use them.