The state of Vibe Coding in enterprise

An exploration of how Generative AI is transforming enterprise software development, examining current platforms, adoption patterns, and the critical gaps that organizations must address to harness this revolutionary approach safely and effectively.

NEKOD

Prepared by

VICTOR MONDAL
Business Development

ANTIGONI KOUROU
Co-founder & CEO

Table of contents

Executive Summary

Generative AI is reshaping enterprise software development through a new paradigm: *vibe coding*. By using natural language to generate code, business and technical users alike can move from idea to execution with unprecedented speed. This shift expands the pool of software builders, but also introduces critical governance and compliance risks.

Our four key insights for enterprise leaders:

1. Vibe coding builds on the low-code paradigm, extending its accessibility while removing the structural constraints of templates and drag-and-drop interfaces.

2. Established players (Microsoft, Google, UiPath, OutSystems) are embedding AI into enterprise platforms, while new entrants (Lovable, Bolt.new) show explosive growth.

3. Enterprises are already capturing value: Walmart saved 4M developer hours in 2024, Citigroup accelerated legacy modernization, and Cognizant scaled AI literacy via 250,000+ employees in global hackathons.

4. The benefits come with risks. Current governance frameworks are built for centralized IT and are not fit for decentralized AI-driven development.

Enterprises that succeed will combine adoption with adaptive governance. Guardrails must "follow the builder" with embedded oversight, real-time compliance, and AI-aware controls. Those who fail risk innovation bottlenecks on one side, or systemic exposure on the other.

What is **"Vibe Coding"** in enterprise context?

Vibe coding refers to using generative AI to build software from natural-language input, rather than writing traditional code line-by-line. The term was popularized by AI pioneer Andrej Karpathy (former Director of AI at Tesla and founding member of OpenAI) to describe how large language models can create executable code from a high-level prompt. In essence, vibe coding flips development to an “intent-first, syntax-second” approach.

Business users or developers simply describe what they need (the “vibe”) by interacting with AI through conversation, describing functionality, workflows, and desired outcomes. The AI model then generates, modifies, and tests the code in real time.

In enterprises, vibe coding:

- Accelerates delivery of internal tools and automations.
- Empowers non-technical teams to innovate directly.
- Bridges IT and business needs without long development cycles.

Platforms enabling Vibe Coding in enterprise

Vibe coding is entering the enterprise through a new wave of platforms. Some are designed for professional developers as AI coding assistants, while others empower non-technical employees, in roles like sales, marketing, and operations, to build their own applications.

This shift beyond engineering teams marks a fundamental change in how software is created inside organizations. In this report, we focus on the platforms specifically targeting non-technical builders.

AI-driven low-code platforms

Unlike traditional low-code/no-code tools, vibe coding removes the need for pre-built templates or drag-and-drop interfaces. It offers flexibility closer to full-code development, but with an AI partner handling the complexity. Let's take a look at the leading platforms.

Microsoft Power Platform

Microsoft is the first mover in the space of vibe coding in enterprise, which started a few years ago with their support for Citizen Development. Microsoft Power Apps is currently undergoing a transformation, from a low-code builder into a full-fledged vibe coding platform that blends generative AI with enterprise-grade governance.

See Microsoft's launch

According to Microsoft, Power Apps now supports agent-first app development, where users collaborate with AI "agents" at every stage—defining user stories, shaping processes, designing logic, and even generating app pages in real code. In contrast to traditional low-code tools' rigid drag-and-drop interfaces, this new approach enables greater creativity and customization.

All the leaders highlighted in Gartner's Enterprise LCAP Magic Quadrant are repositioning themselves as AI-driven app builders. OutSystems and Mendix, for example, are embedding generative AI into their platforms: OutSystems' new AI Mentor can propose and generate app modules from natural language prompts, while Mendix is introducing AI-assisted modeling. Similarly, RPA platforms like UiPath now let users create automation scripts simply by describing the process in plain English.

Even Google has joined in with Duet AI for AppSheet, enabling business users to "describe the app they need and have AppSheet create it" as a starting point. All these solutions share a goal: let non-technical domain experts build software with natural language instead of programming, bringing the speed of no-code together with the flexibility of AI generation.

> *AI-infused low-code platforms like Microsoft Power Platform, OutSystems, UiPath, and Google AppSheet are transforming natural language into enterprise-grade applications; supporting Gartner's forecast that 70% of new enterprise apps will be built this way by 2025. — Gartner via AIMultiple.*

AI-native vibe coding platforms

New players

A new generation of "prompt-to-app" platforms that let non-engineers build applications by conversing with an AI, has emerged. One of the most visible examples is Lovable.dev, a Swedish startup that enables users to describe an idea in natural language, and its AI agent automatically generates the code, designs the interface, and even deploys the app live. Behind the scenes, it taps into advanced models like OpenAI GPT-5, Google Gemini, and Anthropic Claude to generate and deploy code automatically.

Lovable's traction has been explosive: in only 2 months the startup grew from **Zero to $10M ARR** and it has attracted thousands of users, who now build more than 25,000 new products daily using its AI "GPT Engineer" assistant. The appeal is clear: no coding knowledge required. Just describe features in plain language, AI Agents take care of the rest.

Lovable is not alone. Other "prompt-to-app" builders such as Bolt.new follow a similar model, enabling users to spin up full-stack web or mobile apps directly from natural language dialogue in the browser.

Notably, Microsoft has entered this competition as well, by focusing entirely on enterprise. It recently unveiled GitHub Spark, an AI-driven full-stack application builder that can generate a complete web or mobile app from a simple prompt. Spark combines Copilot's code generation with **Azure cloud hosting** and DevOps automation, allowing developers (and even power users) to go from idea to deployed application in minutes. Microsoft claims Spark outputs are production-grade and meet enterprise security standards, with code pushed to GitHub for auditing and version control.

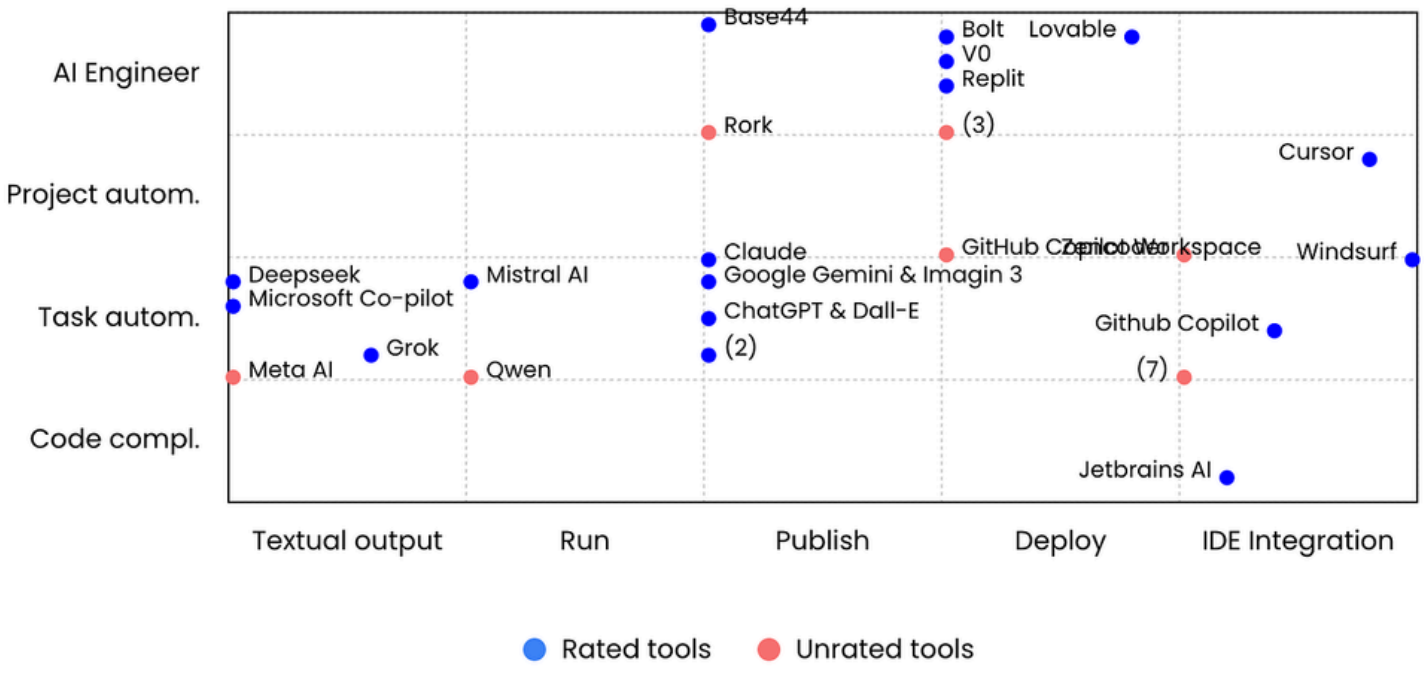
> *AI-native vibe coding platforms are not just accelerating app creation, they're rewriting who gets to innovate, how fast ideas become reality, and how digital power is distributed across the enterprise.*

Agentic AI for code generation

Another class of vibe coding tool blurs the line between coding assistant and autonomous agent. For example, **Replit** (a popular online coding environment) introduced an AI agent that attempts to autonomously build and refine apps for the user. Similarly, startups like **Cursor** and **Windsurf** integrate AI deeply into IDEs (Visual Studio Code, JetBrains, etc.), offering features like "AI Flows" or "Cascade" where the AI can handle multi-step coding tasks or even self-debug.

Windsurf markets itself as "the world's most advanced AI coding assistant" with an agentic AI that can execute complex coding goals in the background. These are aimed at accelerating professional developers' work. For instance, a coder can instruct Windsurf's agent to implement a given feature while they oversee it. Such tools are powerful but currently skew toward tech-savvy users. In fact, a recent analysis noted that "platforms like Cursor and Copilot are mostly confined to a world of trained engineers", whereas the newer vibe coding platforms aim to reach non-engineers. Over time, we may see these agentic tools also gain more natural language interfaces to cater to a wider enterprise audience.

The below chart maps the vibe coding platforms by the type of work they automate (vertical axis) and their role in the software lifecycle (horizontal axis). Developer-focused assistants like GitHub Copilot cluster on the right with IDE integration, while prompt-to-app builders such as Lovable, Bolt, and Replit appear toward the center, enabling non-technical users to publish and deploy applications directly through natural language. (Mapping created by Jeroen Egelmeers)



Use-case adoption across industries

Rapid prototyping and internal apps

Across industries, one of the clearest sweet spots for vibe coding is quickly creating non-critical applications, things like demos, internal dashboards, small web apps for a team, etc. Enterprises are increasingly embracing this shift mainly in hackathons and innovation sprints, where vibe coding allows teams to spin up proof-of-concepts in hours instead of days.

For example, an engineer at a software company shared that he "quickly vibe coded a little UI" during a mini-hackathon to visualize a concept. In another case, a developer used Anthropic's Claude to generate a simple webpage, work that would have taken an hour manually was finished in minutes.

> *By lowering the cost of experimentation, companies can validate ideas faster and move only the most promising prototypes into full-scale development.*

Productivity enhancement

Banks have been among the earliest adopters of AI coding tools, using them to tackle the challenge of modernizing legacy systems. In a highly regulated industry like finance, vibe coding is often limited to internal or backend use cases, where AI improves efficiency without exposing sensitive customer data.

Citigroup has deployed generative coding tools to their employees, for instance, rolled out generative coding tools to more than 30,000 developers, using AI to speed up code analysis and convert decades-old systems.

Retailers are also realizing measurable productivity gains. Walmart reported "early wins" with coding assistants and has since expanded access across its development teams. In 2024 alone, Walmart estimated its engineers saved over 4 million hours through AI-powered code completion and generation. The company's leadership has emphasized that automating routine coding tasks frees up developers to focus on building tools that directly support growth and customer experience.

This demonstrates how even in industries where software is not the core product, AI-driven development can unlock major operational benefits.

Employee reskilling & upskilling

Perhaps the most ambitious adoption of vibe coding is emerging in IT services and consulting firms, which are beginning to scale it across their global workforces.

Cognizant's "Vibe Coding Week" in 2025 brought together more than 250,000 employees worldwide, not just engineers, but also staff from HR, Marketing, and Operations, to participate in an AI-powered hackathon. Using platforms like Lovable, Cursor, and Windsurf, employees turned their ideas into working apps and prototypes in just days.

The initiative was positioned as a way to level the playing field, enabling anyone in the company to build software by "just talking to AI." The results spanned from workflow automation tools to client-facing app concepts, and even included an internally built hub (created with vibe coding itself) to coordinate the event. For Cognizant, this wasn't just an experiment; it was a statement about the future of employee-driven innovation: empowering non-technical staff to contribute directly to digital transformation.

Cognizant's case study

Governance Risks and missing layers in this new ecosystem

As vibe coding decentralizes software creation, control shifts away from traditional IT. To remain secure and compliant, enterprises must modernize governance to match the scale and speed of AI-assisted development.

Shadow IT and security blind spots

The democratization of app development has led to a resurgence of shadow IT. Business users now deploy full-stack applications without IT involvement, increasing the risk of unmonitored data flows, unmanaged code, and unauthorized integrations. Incidents like the Tea App data breach and Replit's AI meltdown highlight how inadequate access controls and insecure environments can lead to material financial and reputational damage.

The **Tea App** was designed as a private platform for women to anonymously share their dating experiences, built with trust, anonymity, and safety in mind. But that promise collapsed when a data leak exposed 72,000 images (including ~13,000 IDs and ~59,000 private photos) and 1.1 million messages, triggering class-action lawsuits. This wasn't a sophisticated hack but a basic misconfiguration. While it's unclear if the issue was directly related to vibe coding, experts warn that the speed-first mindset of AI-assisted development can lead to serious security and compliance lapses.

Compliance & Regulatory Challenges

AI-generated applications are increasingly handling sensitive customer data, yet many lack foundational compliance measures. Most fail to meet standards around encryption, auditability, and jurisdictional controls—putting enterprises at risk of violating **GDPR**, **DORA**, **HIPAA**, and upcoming AI-specific regulations such as the **EU AI Act**. As explainability and accountability become regulatory imperatives, oversight must extend beyond traditional IT systems to include AI-driven outputs.

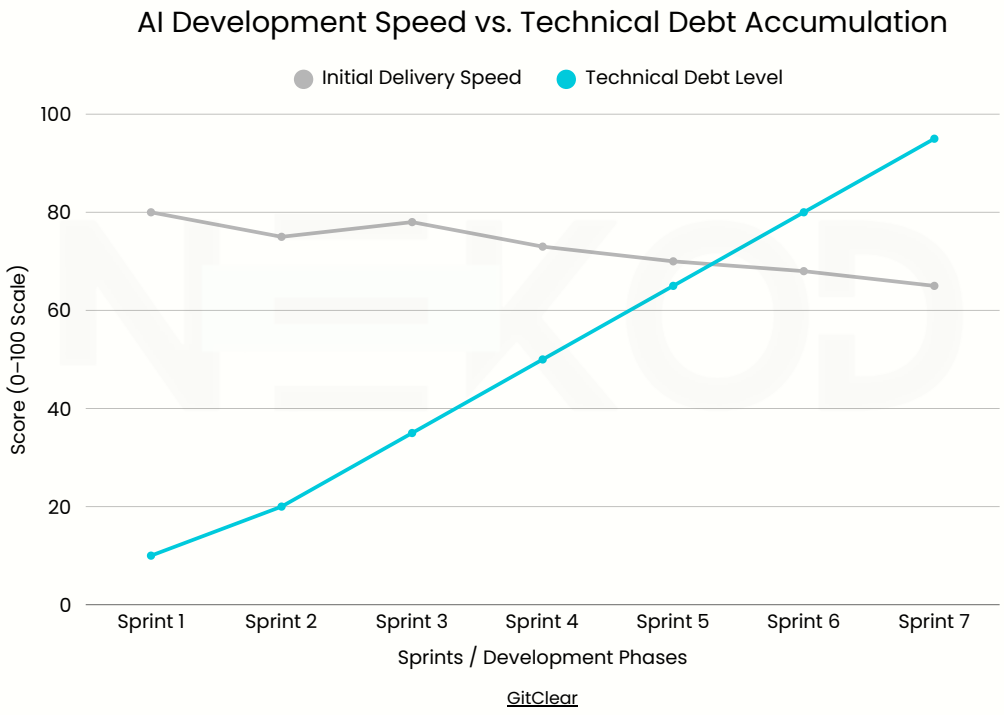
This chart shows how AI-built apps typically fall short of regulatory standards—lacking core controls like audit trails, data residency, explainability, and oversight—creating serious risks from compliance gaps to operational exposure.

Control Area	AI-Built apps (Typical)	Regulatory standard	Risk & Impact
Audit Trail	Not Implemented	Required	No traceability, audit failures, compliance gaps
Data Encryption	Partially Implemented	Required	Risk of data breaches, regulatory fines
Data Residency	Not Implemented	Required	Violates GDPR / cross-border data laws
Model Explainability	Not Implemented	Required (by EU AI Act)	Unclear decision-making, legal non-compliance
Human Oversight	Limited	Required	No real-time control, operational exposure

Compliance Readiness: AI-Coded Apps vs. Regulatory Expectations

Maintainability and Technical Debt

While AI accelerates initial development, it often bypasses architectural planning, resulting in codebases that are brittle, undocumented, and difficult to scale. Enterprises are already encountering fragmented, duplicated logic and increasing technical debt across AI-generated systems. Without standardized review frameworks, knowledge loss and operational risk compound over time—particularly as developers struggle to interpret opaque or hallucinated outputs.



A recent GitClear study revealed concerning trends in AI-assisted development:

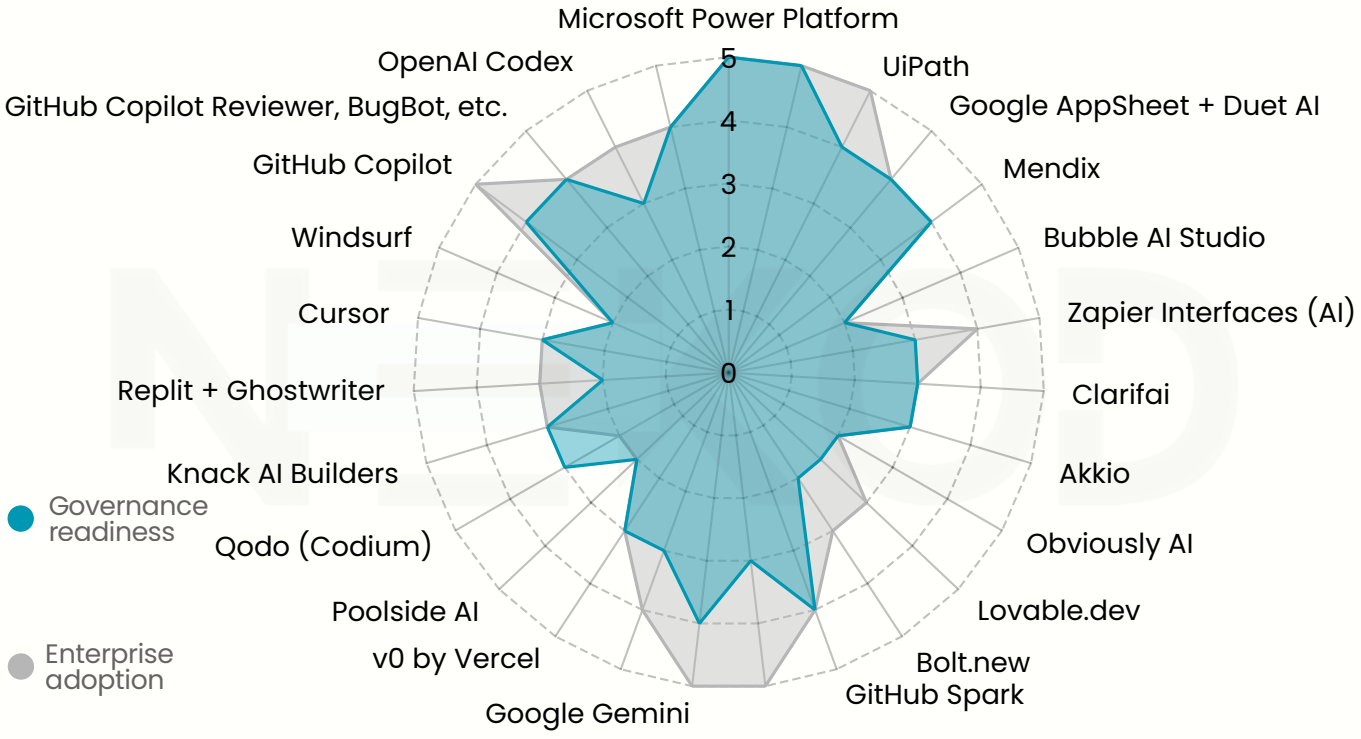
- 41% of junior developers admitted deploying AI-generated code they didn't fully understand.
- 62% said they received no training on reviewing AI code.
- The comment-to-code ratio dropped by more than 50%.
- Refactoring frequency declined by 40%.
- Code duplication surged nearly 8×.

Enterprise gaps in oversight frameworks

Many governance models remain tied to legacy IT assumptions: linear delivery cycles, centralized review boards, and static compliance checklists. These approaches break down when software is developed in minutes by non-technical staff. What's missing is continuous oversight—real-time app registries, policy enforcement engines, and risk visibility that can scale across functions and platforms.

The rise of agentic AI systems will further decentralize development. Governance, therefore, must become embedded, adaptive, and intelligence-driven. Guardrails should follow the builder, not the platform, enabling safe innovation without introducing systemic risk.

Comparative chart of Governance Readiness and Enterprise Adoption across various vibe coding platforms



Our key **takeaways**

The ecosystem of vibe coding platforms is expanding rapidly, spanning everything from AI coding assistants for engineers to prompt-to-app builders for non-technical teams. Two clear clusters are emerging:

1. **Developer-focused tools** (e.g., GitHub Copilot, Cursor, JetBrains AI) emphasize IDE integration and code completion.
2. **Non-engineer platforms** (e.g., Lovable, Bolt, Replit) focus on publishing and deploying

Foundation models like Claude, Gemini, and ChatGPT sit in the middle, powering many of these workflows and extending across categories.

The real disruption lies in tools that move outside of coding environments, empowering business users to create production-ready applications.

The ability to govern at the edge, across platforms, teams, and AI-generated code, will define the enterprise-ready ecosystems of the next decade.

About NEKOD

NEKOD is a governance platform for the new era of vibe coding and AI-built applications. As business users increasingly create apps and automate workflows, enterprises face new risks around security, compliance, and Shadow IT.

Our platform offers:

- Centralized visibility into apps built by business users, outside of engineering teams
- Automated risk and compliance assessments
- Tailored guardrails and proactive policy enforcement

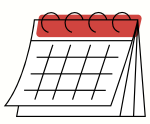
How we help enterprises in practice:

Discover your "Shadow IT" apps with internal audits by piloting with NEKOD

Define and automate your governance framework for AI-driven development

Onboard tools like Microsoft Power Platform across the enterprise with oversight

Empower employees to innovate with curated internal events and hackathons



Ready to embrace AI-driven software?
Schedule a free discovery call here!

NEKOD

August 2025

Thank you!

Report by NEKOD, in collaboration with industry advisors and research from public case studies.

If you have any questions or would like to discuss our findings further, please don't hesitate to reach out.

Amsterdam, Netherlands

LinkedIn

victor@nekod.co

www.nekod.co

Stay up to date with our news.

Follow us on LinkedIn