



AI / AI ENGINEERING / CI/CD

A Practical Roadmap for Adopting Vibe Coding

Vibe coding must be implemented thoughtfully with proper governance to ensure that speed doesn't come at the expense of quality and maintainability.

May 16th, 2025 7:00am by [Emilio Salvador](#)



[GitLab](#) sponsored this post.

A new wave of generative AI tools is redefining the way we build

software and who can participate in the process. At the forefront of this revolution is “**vibe coding**” — using natural language prompts to generate functional code through AI assistance.

Recent industry data shows that nearly half of developers had already integrated AI coding tools by 2023, with vibe coding projects demonstrating **measurable efficiency improvements**. Vibe coding lowers the barriers to entry for development. However, that also leads to lower quality. AI provides the “vibe,” or the suggested pattern, and some **developers might accept it without critical evaluation** or deep comprehension.

Traditional development approaches rely heavily on specific programming languages and syntax rules. Vibe coding lowers the need to comprehend every language and development pattern’s nuances fully, but it does not eliminate that need. This tension between accessibility and quality reflects a broader **transformation in software creation**.

AI is fundamentally **shifting what development means**. Team members can focus on desired outcomes rather than implementation details. Logic, business requirements and user experience precede syntax correctness and language expertise. Organizations increasingly value professionals who can effectively bridge product vision with technical execution — often without writing traditional code.

While vibe coding offers tremendous potential to accelerate development and **democratize software creation**, it must be implemented thoughtfully with proper governance to ensure that speed doesn’t come at the expense of quality and maintainability.



GitLab is the most comprehensive, AI-powered DevSecOps platform for software innovation. GitLab enables organizations to increase developer productivity, improve operational efficiency, reduce security and compliance risk, and accelerate digital transformation.

[Learn More →](#)

THE LATEST FROM GITLAB

[Unlocking AI for every GitLab Premium and Ultimate customer](#)

15 May 2025

[Our step-by-step guide to evaluating runtime security tools](#)

13 May 2025

[Agentic AI guides and resources](#)

7 May 2025

Agentic AI and Vibe Coding

Vibe coding represents an early step in AI-assisted development, and agentic AI furthers this evolution.

Vibe coding is about getting something to appear to work quickly rather than building a robust, efficient and maintainable solution based on solid knowledge. This is where agentic AI can help. Agents can take abstract instructions like “build a customer database” and autonomously handle all the technical implementation details, bridging the gap between quick prototypes and properly engineered solutions.

While **vibe coding primarily focuses on code generation** through

natural language prompts, agentic AI expands these capabilities into an autonomous development ecosystem. This distinction is essential. Vibe coding involves a human developer using AI without requiring deep understanding. Agentic AI involves an AI system taking on a more proactive planning and autonomous role in building software based on a given goal.

The relationship between vibe coding and agentic AI is symbiotic. Vibe coding provides the foundation for human-AI interaction through natural language, while agentic systems build upon this foundation to create self-directed development partners. These intelligent systems respond to prompts and anticipate needs, make independent decisions and take action with minimal supervision.

Agentic AI systems enhance vibe coding by **integrating deeply into development workflows**, conducting sophisticated code reviews, recommending infrastructure optimizations and adapting to changing requirements. **Industry research from Deloitte** indicates that 25% of companies using generative AI will implement agentic AI pilots in 2025, which is expected to double by 2027.

Implementing vibe coding and agentic AI together requires careful planning. Organizations must establish **comprehensive security protocols**, ensure compliance with data regulations, and create clear communication channels between AI systems and existing tools. Despite these implementation challenges, the combined power of vibe coding and agentic AI offers compelling benefits in development speed, code quality and resource optimization.

TRENDING STORIES

1. [Thoughtworks CTO: AI Means We Need Developers More Than Ever](#)
2. [Remote MCP Servers: Inevitable, Not Easy](#)
3. [A Guide To Navigating GPU Rentals and AI Cloud Performance](#)
4. [Why Canva Chose MCP Server Over AI Agent for App Developers](#)
5. [Frontend's Next Evolution: AI-Powered State Management](#)

Taking an Evolutionary Approach to Implementation

Development teams and technical leaders can follow this evolutionary path to effectively implement vibe coding and agentic AI:

1. **Begin with AI assistance:** Introduce [developers to AI tools that improve productivity](#) for routine tasks. Focus on building familiarity, comfort and confidence with AI assistance for coding, documentation and simple problem-solving.
2. **Expand AI assistance across the software development life cycle:** Move beyond just code writing to integrate AI tools into testing, debugging, code review and documentation. Identify repetitive, time-intensive workflows where AI can create immediate value with minimal disruption.
3. **Establish governance frameworks and interoperability standards:** Create clear policies for use of AI tools, including data access permissions, security protocols and quality standards. Define protocols for how AI systems will share information and collaborate across platforms.
4. **Introduce autonomous AI agents for specific tasks:** Deploy agents to handle self-contained development tasks with a degree of autonomy. These agents take abstract goals like "optimize this database query" and handle the implementation details independently while maintaining code quality.

5. **Scale agent implementation across the organization:** Expand the scope of tasks handled by agents and introduce multiple agents working together on complex projects. Integrate agents deeply into the end-to-end software development life cycle and redesign team structures to create cross-functional groups combining technical expertise and domain knowledge.
6. **Continuously improve through feedback and education:** Implement systems to monitor agent performance with clear metrics and correction protocols. Invest in organizationwide AI literacy through training programs for prompt engineering, AI collaboration techniques and effective system oversight.

This evolutionary approach ensures technical implementation and organizational leadership progress together in the AI transformation journey, maximizing the benefits of vibe coding while building robust, efficient solutions.

The Changing Developer Landscape

The engineering role is evolving as vibe coding and agentic AI handle more routine development tasks. Less experienced developers face a steeper learning curve with fewer straightforward tasks available for initial skill-building. Simultaneously, senior engineers must adapt as AI takes over traditional oversight responsibilities.

The industry is witnessing growing demand for new specialized roles like prompt engineers who effectively guide and refine AI outputs. The most valuable skills now include architecture design, strategic thinking and the ability to collaborate with AI systems effectively.

While these shifts may create downward pressure on certain roles and salaries, they also create opportunities for developers who embrace AI as partners rather than threats. The most successful engineers will be those who leverage AI to handle routine tasks while

focusing their expertise on innovation and strategic problem-solving.

Organizations that embrace vibe coding and agentic AI gain significant competitive advantages through accelerated development cycles, improved code quality and more efficient resource allocation. Those who fail to adapt risk being outpaced in an increasingly AI-powered development landscape.

TNS



Emilio Salvador is vice president of strategy and developer relations at GitLab. A technology executive with more than 20 years of experience, Emilio has held roles at Amazon and Microsoft, and most recently led strategy and operations for the Developer...

[Read more from Emilio Salvador →](#)

SHARE THIS STORY



TRENDING STORIES

1. [Thoughtworks CTO: AI Means We Need Developers More Than Ever](#)
2. [Remote MCP Servers: Inevitable, Not Easy](#)
3. [A Guide To Navigating GPU Rentals and AI Cloud Performance](#)
4. [Why Canva Chose MCP Server Over AI Agent for App Developers](#)
5. [Frontend's Next Evolution: AI-Powered State Management](#)

INSIGHTS FROM OUR SPONSOR



GitLab is the most comprehensive, AI-powered DevSecOps platform for software innovation. GitLab enables organizations to increase developer productivity, improve operational efficiency, reduce security and compliance risk, and accelerate digital transformation.

[Learn More →](#)

Unlocking AI for every GitLab Premium and Ultimate customer

15 May 2025

Our step-by-step guide to evaluating runtime security tools

13 May 2025

Agentic AI guides and resources

7 May 2025

Automate tedious coding tasks with GitLab Duo Workflow

6 May 2025

How to use GitLab's Custom Compliance Frameworks in your DevSecOps environment

30 April 2025

GitLab Duo + Amazon Q: Transform ideas into code in minutes

28 April 2025

TNS DAILY NEWSLETTER

**Receive a free roundup of the
most recent TNS articles in
your inbox each day.**

EMAIL ADDRESS

SUBSCRIBE

The New Stack does not sell your information or share it with unaffiliated third parties. By continuing, you agree to our [Terms of Use](#) and [Privacy Policy](#).

ARCHITECTURE

Cloud Native Ecosystem
Containers
Databases
Edge Computing
Infrastructure as Code
Linux
Microservices
Open Source
Networking
Storage

ENGINEERING

AI

AI Engineering
API Management
Backend development
Data
Frontend Development
Large Language Models
Security
Software Development
WebAssembly

OPERATIONS

AI Operations
CI/CD
Cloud Services
DevOps
Kubernetes
Observability
Operations
Platform Engineering

CHANNELS

Podcasts
Ebooks
Events
Newsletter
TNS RSS Feeds

THE NEW STACK

About / Contact
Sponsors

Advertise With Us

Contributions



Community created roadmaps, articles, resources and journeys for developers to help you choose your path and grow in your career.

Frontend Developer Roadmap

Backend Developer Roadmap

Devops Roadmap

FOLLOW TNS



© The New Stack 2025

[Disclosures](#)

[Terms of Use](#)

[Advertising Terms & Conditions](#)

[Privacy Policy](#)

[Cookie Policy](#)