



Q2 2025

# TECH TRENDS REPORT

dexian®

# The Rise of AI-Driven Development Frameworks



In March 2025, [OpenAI CEO Sam Altman](#) had this to say about the impact of artificial intelligence (AI) on software development:

*"My basic assumption is that each software engineer will just do much, much more for a while. And then at some point, yeah, maybe we do need less [sic] software engineers."*

He's not alone in his perception. [Microsoft](#), [Meta](#), other big tech companies – and even small organizations – are rethinking their software development processes, evolving from traditional coding to AI-assisted frameworks. Though we still think there's a place for human coders in the software development lifecycle, ethical organizations can accelerate delivery and the performance of coding teams with AI. This report explores:

- Where AI Is Making the Biggest Impact
- Essential Tools Powering AI-Driven Development
- How AI Is Reshaping Enterprise IT Strategy
- Common AI-Coding Pitfalls and How to Avoid Them
- What Dexian's AI-First Approach Looks Like in Action

Let's explore the world of AI-driven development.



# Where AI Is Making the Biggest Impact



The well-defined rules, clear syntax, and measurable outputs make code writing a perfect fit for AI. Moreover, large language models (LLMs) and AI tools can be used throughout the entire software development lifecycle (SDLC) to accelerate the delivery of accurate and readable code.

The primary use cases gaining traction among AI-savvy developers include:

## CODE SUGGESTIONS

This is often the first thing that comes to mind when discussing AI-driven development. Whether trained on a company's code base or open-source repositories, code suggestion tools can recommend the next line of code or even produce full functions. Since these tools are context aware (attuned to the surrounding code and file structures), they can make exceptional suggestions. Plus, they're perfect for routine or repetitive code writing – the sort of tasks that would have developers watching the clock, not breaking a sweat.

## AUTOMATED TEST GENERATION

Writing tests is essential for maintaining code quality, but it's often time-consuming and repetitive. AI-powered tools can automatically generate unit tests, integration tests, or even edge case scenarios based on the existing codebase. These tools not only save time, but also catch potential issues that developers might overlook, helping teams maintain higher standards without slowing down the development cycle. It's like having a second set of eyes on your code (only it's faster, tireless, and built for coverage).

## CHATBOT FUNCTIONALITY

When developers had quick questions, they used to wade through online forums in the hopes of finding the perfect answer. Now AI coding assistants can respond to queries and offer suggestions, sometimes directly within your interactive development environment (IDE). This instant support keeps developers focused and productive, offering help with everything from debugging to understanding unfamiliar code.

## CROSS-LANGUAGE TRANSLATION

Are you migrating legacy tools in outdated languages? Are your developers trying to collaborate across programming languages? AI tools can help translate code from one programming language to another, preserving functionality while adapting the preferred language's syntax and conventions.



# Essential Tools Powering AI-Driven Development



Since the “Big Bang” of generative AI introduced coding assistants to the world, we’ve seen an explosion of options on the market. Though some of the big contenders are ideal for complex and context-heavy scenarios, the smaller platforms also show some versatility and specialization within specific stages of the software development lifecycle.

## **GitHub Copilot**

This commonly used tool is the brainchild of GitHub, OpenAI, and Microsoft. Copilot offers code generation, a chatbot for your queries, and integrates seamlessly with various languages and IDEs. However, because Copilot is trained on public code repositories, there are privacy concerns about including copyrighted or sensitive code.

## **Amazon CodeWhisperer**

If your team uses Amazon Web Services (AWS), then Amazon CodeWhisperer might be familiar to you. This AI coding assistant makes real-time code suggestions based on your current and previous inputs, conducts security scans to detect vulnerabilities, and monitors unwanted similarities to open-source code.

## **Google Gemini Code Assist**

Powered by Gemini’s AI model, this AI coding tool is useful for writing, testing, debugging, and documenting code quickly and efficiently. An increasing number of plug-ins in VS Code, JetBrains, and other environments reduce user friction and accelerate delivery. Furthermore, Code Assist is designed with observability and security in mind – perfect for ethical implementation.

## **Windsurf (Codeium)**

This AI development tool is quickly growing in adoption. Windsurf supports 70+ programming languages with its AI-powered autocomplete, chatbot features, debugging, and documentation. There are some limitations to indexing and context awareness in the free version, but the paid version closes that gap.

Though the above tools address many AI development needs, there are instances when niche software or emerging tools are more attractive. The multitude of options out there means you can find an AI tool that fits your precise needs, but you need to ensure that your vendor is:

- **Ideal for the SDLC stage you want to automate**
- **Compatible with your preferred IDE and programming languages**
- **Focused on security and privacy**

# How AI Is Reshaping Enterprise IT Strategy



Below are some of the ways AI-driven development will be a catalyst for even bigger change.

### Enhanced Productivity

AI-powered coding assistants don't just offer a speed boost. These tools take on repetitive tasks and accelerate development cycles, allowing teams to push applications into production faster and with greater precision. The result? Faster release cycles that still allow time for testing, debugging, and other quality control measures.

### Cost Efficiency

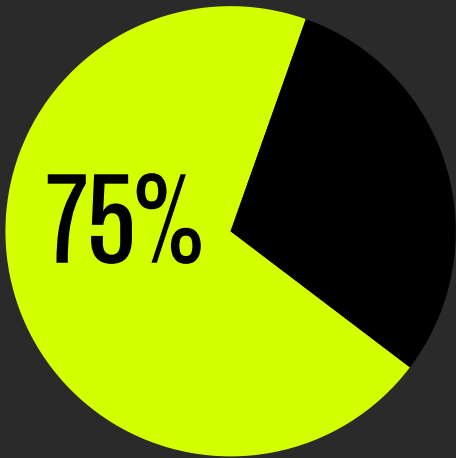
Every hour saved is a dollar (actually, many dollars) earned. By minimizing manual coding and streamlining the development process, AI tools help reduce overhead and optimize resource allocation. If used correctly, this software can reduce the cost per project, allowing companies to reinvest their savings into innovation and long-term growth.

### Skill Transformation

Today's teams need fluency in AI-driven workflows as much as traditional languages and frameworks. However, there's a fine line between augmenting your people and turning them into factotums for AI coders. Organizations that proactively upskill their talent are positioning themselves to lead, not lag, as the nature of software development continues to shift.

### Security Considerations

Speed is valuable but only when it's secure. As AI-generated code becomes more prevalent, enterprises must strengthen security protocols to catch issues early. That means more robust testing, smarter DevSecOps integration, and a mindset that treats AI as a partner, not a shortcut.



75% of Software Engineers Will Use AI Code Assistants by 2028



# Common AI-Coding Pitfalls and How to Avoid Them



Like any technology, there's a massive difference between smart and slipshod AI deployment. First and foremost, monitor how AI tools are impacting your workforce. Though code assistants can boost productivity, they can also [make software engineers feel](#) like their work is "more routine, less thoughtful, and much faster paced." There needs to be a balance between maintaining a sustainable headcount and implementing AI to achieve heightened output without hurting morale.

Organizations also need to moderate an emerging trend in software development: vibe coding. This is what happens when developers allow AI to generate code without oversight or testing. At its worst, people vibe code without understanding a single line of the output. This can result in broken features, mandatory refactoring in the future, and even cybersecurity risks such as [package confusion attacks](#). Encouraging your developers to take responsibility for AI-generated code can mitigate some risks.

AI coding tools are only as good as their training data. They'll replicate insecure patterns or produce subtly flawed code. And since AI chatbots make recommendations with such confidence, an inexperienced developer might accept errors without a second thought. The best way to counteract this risk is to encourage skepticism from junior developers as well as experienced team members. When collaboration is common, developers can rely on each other for additional quality checks.

At the end of the day, coders are liable for any programming output. Your team is on the hook for any AI-generated code that reinforces biases, entrenches vulnerabilities, or damages functionality. So, it's important to cultivate a culture of accountability, where your team validates code quality before it is productionized and is proactive about identifying bugs.

# What Dexian's AI-First Approach Looks Like in Action

In practice, embedding AI across the IT lifecycle results in both operational efficiencies and strategic gains. Yet without an experienced partner to help with the selection, implementation, and support of AI-driven development tools, your organization might struggle to elicit the full ROI.

Working with Dexian IT Solutions can simplify the AI implementation process. Our extensive experience with automation and AI, as well as our expansive partnerships with numerous IT vendors, can help your organization to pinpoint the perfect coding assistants and AI tools for your team.

The results? Dexian's delivery framework demonstrates this impact across multiple client environments and creates observed benefits:

## KEY TAKEAWAYS FOR Q2 2025

**AI-driven development** is accelerating software release cycles and improving code quality.

The **right tools can** reduce planning time, boost productivity, and speed up testing.

**Security risks rise** when AI code isn't reviewed or understood by developers.

**Upskilling teams** in AI workflows is essential for long-term success.



Reduces Planning Cycles by **35%**



Increases Developer Productivity by **45%**



Accelerates Regression Testing by **60%**

Best of all, you can make this progress while keeping developers on your payroll. We're all committed to finding the perfect balance between people and AI programming.

Learn how to ethically harness AI-driven development. **Partner with Dexian today.**

EXPLORE OUR ARTIFICIAL INTELLIGENCE SOLUTIONS

dexian®