inclusioncloud
DIGITAL ENGINEERING

EDITION 3

The CIO's Strategic Guide for 2025

Rethinking Tech Hiring for the AI-Augmented Era

Axis / The pivot point for business strategy.

Engineered by InclusionCloud.

Executive Summary

Welcome to the third edition of AXIS, the Inclusion Cloud research lab. This time, we present Rethinking Tech Hiring for the AI-Augmented Era—a report that explores how AI is rewriting the rules for identifying and hiring elite tech talent.

In a world where almost anyone can generate working code in seconds with the help of AI, the usual hiring signals are no longer enough. This shift is creating a domino effect across the entire hiring journey and brings one critical question to the forefront:

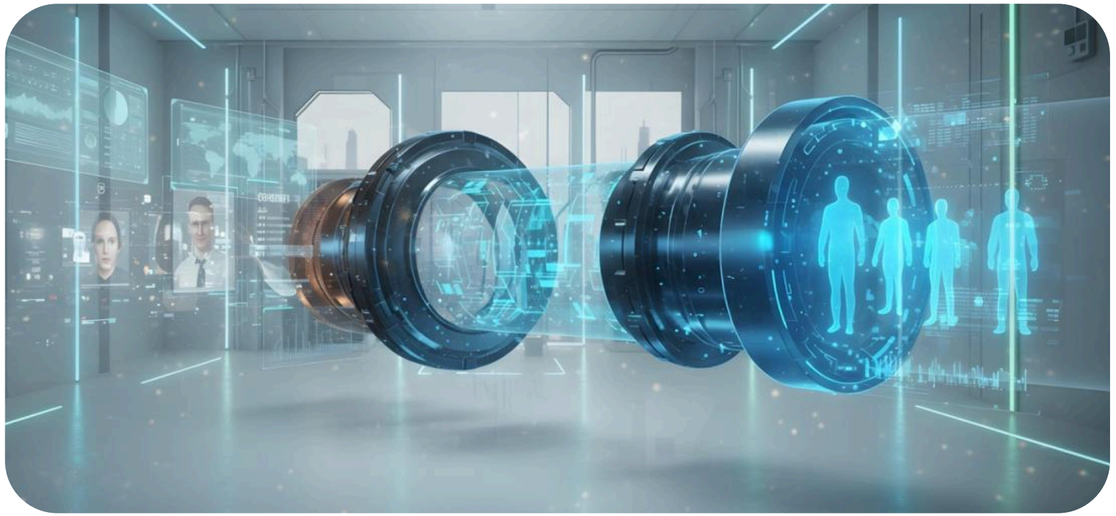
If anyone can produce code that seems functional at first glance, how do you tell apart those who truly understand architecture and systems from those who simply prompt, copy, and paste without context or long-term thinking?

This report introduces the concept of vibe coding—what it is, how it's spreading, and whether it's leading us toward something even riskier: vibe architecture.

Let's be clear: we aren't against the use of AI for software development. Quite the opposite.

We see AI as a powerful productivity accelerator when used by the right people in the right way. Our goal with this report is to highlight the growing divide between developers who use AI as a strategic extension of their expertise (the AI-augmented developers) and those who rely on it as a workaround for skill gaps (the vibe coders).

We also explore how Inclusion Cloud is redesigning its recruiting engine to meet this new reality.



inclusioncloud

IA Is Taking Software Development by Storm

The way we connect with customers is changing. The way we search online is changing. The way we access knowledge is changing. And of course, the way we build software is changing too.
GitHub's AI in Software Development report revealed that 97% of software engineers, developers, and programmers have already used AI in their work. Even more interesting, 71% say AI helps them overcome barriers when learning new programming languages or trying to understand existing codebases.

Google is another clear example. CEO Sundar Pichai recently shared that the company is accelerating its coding process with AI, and 25% of all new code is already being generated with Gemini.
So yes, all signs indicate that AI is here to stay in the software development lifecycle. But there's something important we need to clarify.

Writing code is only one part of the software production chain. It matters, but so do many other stages before and after development itself.

You can think of it like this: writing code is like composing a melody. Building software is like producing an entire album.

There's the definition of the problem. System architecture and technology selection. Release planning, feature validation, automated testing, security practices, system integration, and of course, maintaining and evolving the product once it's live.

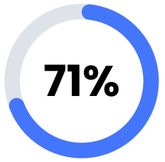
Each of these stages demands decisions, context, and coordination. Without that, even the best AI-generated code falls short. Even in Google all the AI-generated code is still reviewed and accepted by their engineers.

That's why we don't believe AI is coming for developers. We believe it will work alongside them, because much of this chain still relies on human capability, experience, and vision. But while AI is transforming how software is built, it's also reshaping how we assess those who build it. And that's where the next challenge begins.

How AI Is Reshaping Software Development

Key Insights:

of developers already use AI in their workflows.



say AI helps them learn new languages and understand legacy code.

of Google's new code is AI-generated

Gemini

PLAN BUILD TEST RELEASE MAINTAIN

MOST OF THE SOFTWARE LIFECYCLE STILL DEPENDS ON HUMAN EXPERTISE.

1.

The New Hiring Challenge:

Everyone Looks Like a Developer

AI tools have transformed what a candidate can show in an interview. With the help of coding assistants, many professionals—including those with limited experience—can now generate working code, solve small tasks, and present clean outputs that meet surface-level expectations.

In that context, the line between someone who "knows how to code" and someone who "understands software development" has become harder to define.

At first glance, the distinction seems minimal. But when the code is reviewed more carefully—or deployed into a real system—the differences become clear. Code that runs is not necessarily code that scales. A function that returns the right value does not guarantee that the solution fits the overall architecture, handles edge cases, or supports the business logic behind the feature.
This is where traditional hiring processes begin to fall short.

For years, evaluating technical talent was tied to visible outputs: working code, passing tests, confident syntax. But AI has blurred the line between demonstration and depth. It's now entirely possible for a candidate to pass early-stage screenings by relying on AI to write code they do not fully understand.

In our recent Inclusion Cloud poll, 57 percent of developers said AI makes them significantly faster—but only for certain types of tasks.

That nuance matters. It reflects what technical leads are seeing on the ground: speed does not always mean understanding, and output does not always mean ownership.

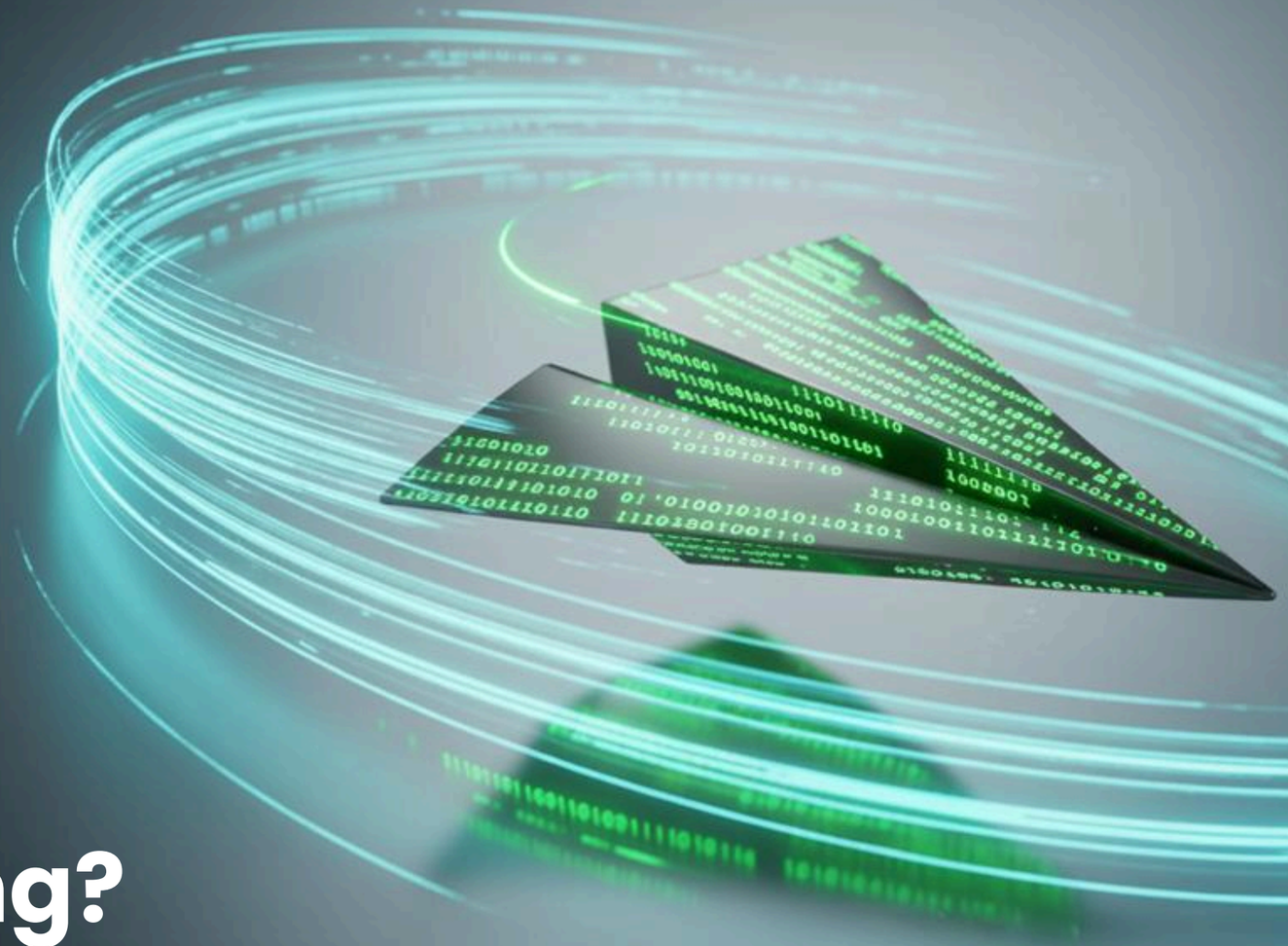
Candidates who succeed in isolated exercises often struggle when asked to explain trade-offs, align their choices with long-term maintainability, or contribute to larger systems with multiple dependencies. There is a growing gap between apparent fluency and architectural thinking.

At Inclusion Cloud, we refer to this phenomenon as the new hiring challenge. **It is no longer about identifying who can write code. It is about identifying who can build software that performs, integrates, evolves, and supports business objectives over time.**

This shift demands a change in how we screen and select candidates. It requires looking beyond working outputs to understand the reasoning, the decision-making, and the depth behind every solution.

More than ever, it is critical to draw a clear line between an AI-augmented developer—someone with real experience, architectural thinking, and a strong foundation in computer science—and a vibe coder, who relies on copy-paste code without understanding how it works or whether it will hold up under pressure.

inclusioncloud



2.

What Is Vibe Coding?

And Why It Matters

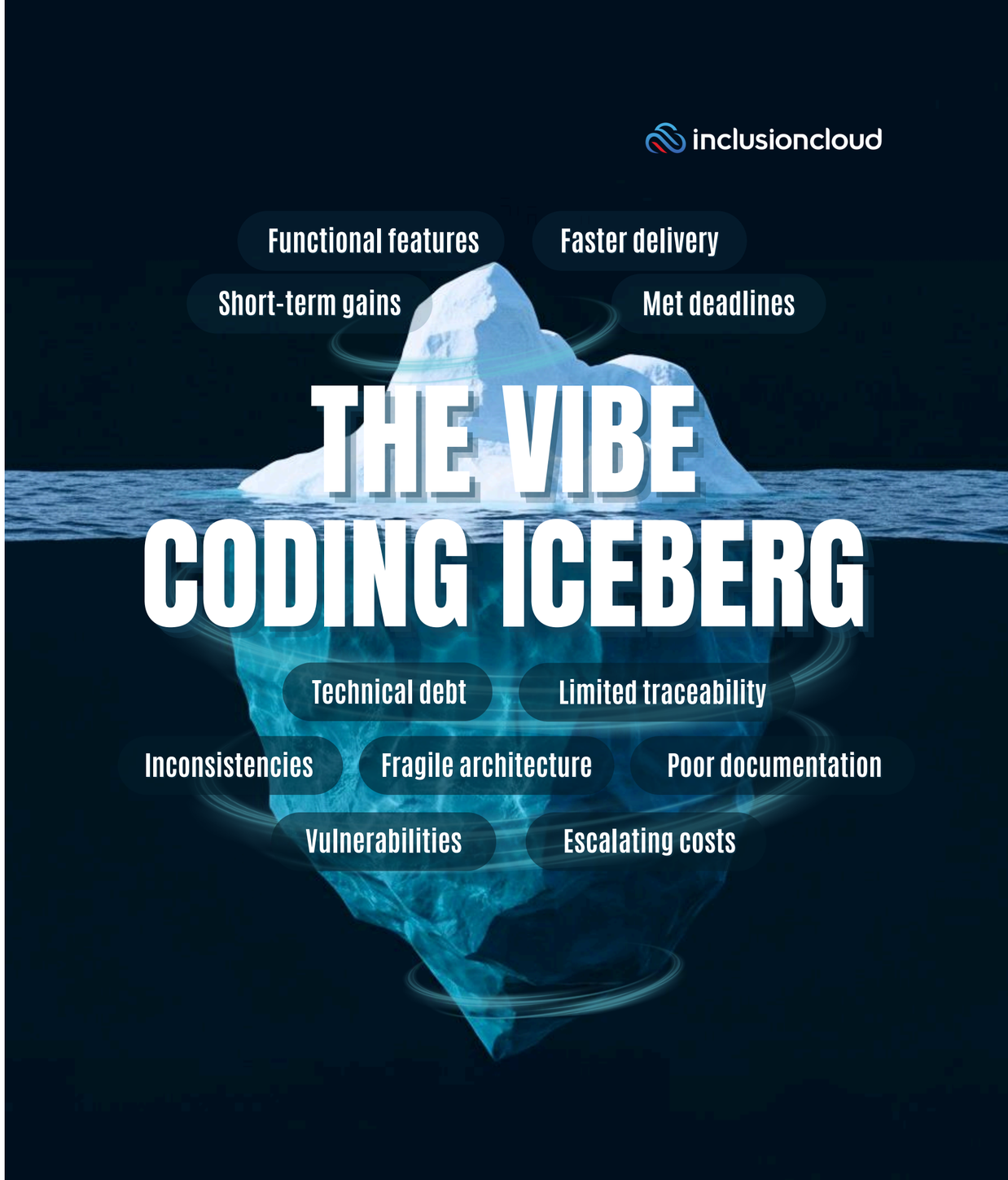
Vibe coding refers to the practice of prompting an AI model like ChatGPT, Claude, or GitHub Copilot to write code without having a deep understanding of the task or carefully reviewing the results.

The typical approach is to copy and paste the generated code and see if it works. If it doesn't, the user simply re-prompt the model instead of debugging or analyzing the issue.

This is not how all developers use AI. What we call AI-augmented developers take a very different approach: they use these tools thoughtfully—reviewing, iterating, understanding the generated code, and considering how it fits into the broader architecture of the project.

In our recent poll, 33% of respondents said the real trade-off with vibe coding it's often about shipping fast and fixing later. That might work in prototyping or low-risk environments, but for enterprise-grade software, the consequences can be costly.

The risk is that this casual coding culture leads to vibe architecture: systems built with patchwork logic, poor documentation, and hidden technical debt.



inclusioncloud

3.

The Role of the AI-Augmented Developer

The developers leading this new chapter of software development don't just use AI. They know how to integrate it into the broader engineering process with intention and clarity.

We call them AI-augmented developers.

These are senior professionals with strong technical foundations. Many have studied computer science, worked on complex systems, and earned certifications in platforms like Salesforce, ServiceNow, or SAP. They understand how different components interact, how to weigh trade-offs, and how to design systems that can grow and adapt over time.

They rely on AI to speed up routine tasks. That gives them more room to focus on what matters most: software architecture, performance, scalability, and aligning with the business logic behind the product.

In our latest Inclusion Cloud poll, 74% of tech professionals said the impact of AI on code quality depends on how it is used. This reinforces the importance of assessing more than just working code or delivery speed. Recruiters must look for sound engineering judgment, problem-solving ability, communication skills, and teamwork.

AI-augmented developers often work across departments:

They collaborate with product teams, designers, and stakeholders. Their ability to explain technical needs or challenges clearly is becoming one of the most valuable skills in modern development teams, especially now that AI tools can produce code in seconds.

These developers **know how to refactor and validate AI-generated code. They think critically, connect what they build with the rest of the system, and ensure it aligns with long-term goals.** For them, AI is not a shortcut. It is a starting point.

GitHub found that between 60% and 75% of developers using AI feel more fulfilled, less frustrated, and more engaged in meaningful work. Stack Overflow adds that 81% of developers see productivity as the main benefit of AI. For those just learning to code, 71% say AI helps them accelerate that process. But among experienced developers, only 61% say the same.

That difference reveals something important. **For senior developers, the biggest value of AI isn't learning. It's the ability to reclaim time, reduce backlog pressure, and focus on the parts of the job that have the most visible impact. These are the moments where creativity and system thinking shine. And these are the developers who will define the future of software.**

Reclaiming Time, Refocusing Talent

inclusioncloud

Key Stats:

74%

of tech professionals say AI's impact on code quality depends on how it's used.

– Inclusion Cloud Poll, 2025

60-75%

of developers using AI feel more fulfilled, less frustrated, and more engaged.

– GitHub's AI in Software Development Report

81%

of developers cite productivity as the top benefit of using AI.

– Stack Overflow Developer Survey

71%

of early-career developers say AI helps them learn faster— but only **61%** of experienced developers say the same.

– Stack Overflow Developer Survey

inclusioncloud

4.
AI is here to stay
and that’s a good thing
(if used right)

Let’s be clear: there’s no turning back. AI is already embedded in the daily workflow of millions of developers, and that adoption curve is only accelerating.

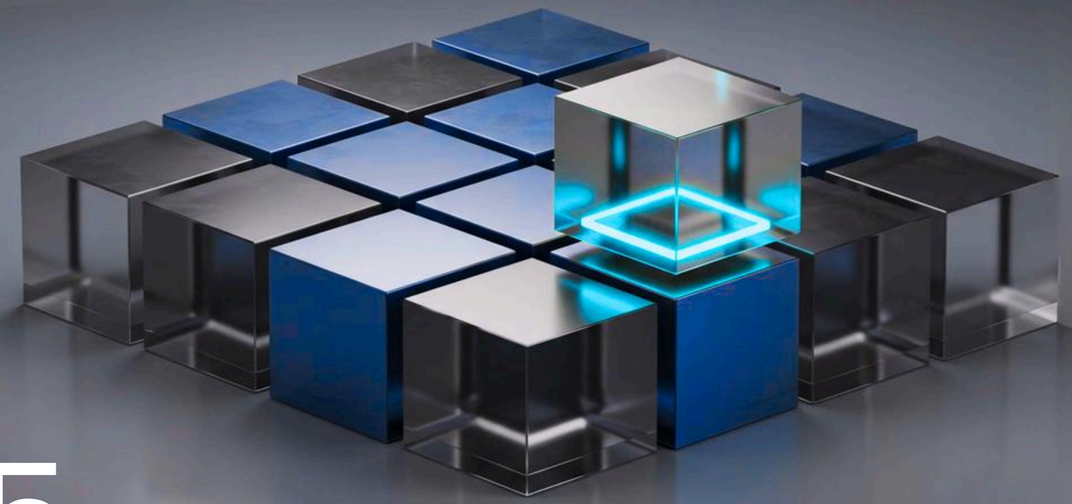
At Inclusion Cloud, we don’t believe AI should be limited or avoided. When used with intention and expertise, it becomes one of the most powerful accelerators of productivity in decades.

That’s why we’ve embraced AI—not just as a tool, but as a shift in how software is built, how teams operate, and how talent must be evaluated. And just like technical fundamentals or soft skills, we believe AI fluency will soon become a standard part of the hiring process.

But having access to AI isn’t what makes the difference. The real differentiator lies in how developers use it. Because that’s what reveals who’s building real value—and who’s just generating passable outputs.

This shift is already in motion. And companies that fail to adapt their hiring processes won’t fall behind because they lack AI tools, but because they lack the right people to use them wisely.

AI is everywhere. What sets top teams apart is how they use it:



5.

Two Types of Developers in the AI Era

As AI becomes a central part of software development, we've observed a clear distinction emerging between two types of developers. They might both use AI—but what they do with it is very different. And so is the way you should assess them.

To break this down, we spoke with **Pablo Bucci,** Recruiting Lead at Inclusion Cloud, who has spent the last two years building new methodologies to identify and validate developers in the AI era.

> ***"Not all AI usage is the same," Pablo explains. "Some developers use AI within enterprise systems where the logic is predefined. Others use generative AI to write custom code. And that has a huge impact on how you evaluate their skills."***

Let's look at each group.

inclusioncloud

A.
Developers in Low-Code Platforms with Embedded AI

These developers work within enterprise ecosystems like ServiceNow, Oracle, SAP, or Salesforce. Most of their output relies on visual configuration, pre-built components, and process automation. The AI is integrated into the platform to optimize delivery and standardize workflows.

"In these environments, a lot of the complexity is abstracted," Pablo notes. "So you need to ask: what did the candidate build? Why did they build it that way? Did they improve the workflow or just configure what was asked?"

Here, hiring is more straightforward. Certifications, use-case experience, and platform familiarity serve as concrete benchmarks.

But Pablo cautions that it's still important to go deeper. "When we're hiring for these roles, we want to know if the candidate understands the business logic behind the automation. Because that's how you separate someone who can truly add value from someone who's just completing steps."

B.

Developers Using Generative AI for Custom Code

This second category includes developers who use AI tools like GitHub Copilot, ChatGPT, or Claude to write backend logic, frontend components, or full-stack features. Unlike the low-code environment, this group prompts the AI directly—and how they do it reveals their level of depth.

“This is where it gets tricky,” Pablo says. “We’ve had interviews where the code looks clean, but falls apart under real questions. Or where candidates clearly rely on AI-generated answers without fully understanding what’s behind them.”

And that’s a crucial point: GenAI can help candidates produce clean-looking code, but that doesn’t mean they understand it. Writing code with AI isn’t the same as owning the logic behind it.

That’s why Pablo’s team has shifted their approach. They use real-world challenges and contextual questions to understand how candidates think, how they troubleshoot, and whether they can explain architectural decisions.

“Anyone can generate working code. But the real value is in understanding why that code was written that way, what trade-offs were made, and how it fits into the bigger system.”

— Pablo Bucci, Recruiting Lead at Inclusion Cloud

inclusioncloud

6.

How We Built a Recruiting Engine for the AI Era

We didn’t wait for the AI wave to crash over us; we moved first. We built our recruiting engine around it.

From the beginning, our goal was to build better solutions for the new era of AI-augmented development. We realized that AI could help us do this more effectively, but only if paired with human expertise. That’s why we designed a two-layer engine: one powered by AI to scale, and one powered by people to refine and ensure depth.

- In the first stage, **we built an AI-powered engine to identify top talent from our database** of over one million candidates. It analyzes everything from resumes to GitHub activity and LinkedIn profiles, surfacing developers with the right skills, experience, and adaptability for each role.

- In the second stage, we expanded our criteria. **We now evaluate how each developer uses AI—not just whether they’ve used it, but how they apply it in real-world settings.** Do they understand prompt engineering? Can they refine AI outputs to meet architectural and security standards? Have they used AI to accelerate their workflow, or are they just copying and pasting?

Once a candidate matches, our team steps in. Our recruiters conduct context-first interviews—asking how candidates made key decisions, collaborated with others, or approached complex business logic.

We don’t just look for lines of code. We want to know how they think, how they solve real problems, how they explain trade-offs, and how they adapt to changing requirements.

- **Then we move to the technical deep dive, led by our own senior engineers.**

  These interviews aren’t scripted. They’re grounded in realistic challenges. We ask about scaling systems, managing legacy migrations, or solving API performance issues. We evaluate how candidates reason through problems, communicate across teams, and apply standards to unpredictable scenarios.

And we always ask: **how do you use AI in your process?**

We want specifics. Which models do you prefer—GPT, Claude, open source? Why? Can you walk us through how you’d prompt a model to build a pipeline, or how you’d review an AI-generated Terraform script before using it in production?

Their answers reveal more than AI knowledge. They show how candidates manage risk, apply system thinking, and decide what still needs human oversight.

inclusioncloud

How the Inclusion Cloud Team Identify AI-Augmented Developers

Ask for real examples

We ask about specific scenarios. What problem were they solving? How did they choose that approach? What would they do differently today?

Watch body language

We always ask for the camera to be on. You can often tell when someone is reading from a second screen or stalling.

Cross-check their story

We compare their answers to what's on GitHub and LinkedIn. If their claims don't match their portfolio, that's a red flag.

Test communication and ownership

We use situational questions like, 'What would you do if a delay came from another team?' It shows how they handle collaboration with stakeholders.

Run live code sessions

We observe how they think under pressure. If they rely too heavily on search or pause to read AI output, we can tell.

Use senior engineers for final validation

Every candidate goes through a second round with our engineers. We recreate real project dynamics and see how they respond.

Look for depth

Generic solutions usually mean the candidate didn't write or understand the code. We dig into their reasoning and architectural decisions.

7.

What’s Next: Rethinking Your Talent Strategy

inclusioncloud

AI-Era Talent Strategy:

5 Key Moves

Based on our findings, here's what we recommend for companies navigating AI-era hiring.

HIRE FOR THINKING, NOT JUST CODING

Reveal how candidates solve problems, reason through complexity, and align technical decisions with business goals.

PRIORITIZE SENIOR ENGINEERS

Anchor teams with certified, experienced talent to ensure architectural integrity, mentoring, and long-term system quality.

SCREEN FOR AI FLUENCY

Evaluate how candidates use AI—how they prompt, validate outputs, integrate results, and understand model behavior.

TEST IN REAL-WORLD CONDITIONS

Use interviews that simulate system constraints and architectural trade-offs to evaluate critical thinking under pressure.

ACCEPT AI, EXPECT RESPONSIBILITY

Assume AI is in use. Focus on how candidates apply it wisely, securely, and with accountability.

Final Thought

Throughout this AXIS report, we've explored how AI is transforming software development—not just in how code is written, but in how teams are structured, how talent is evaluated, and how businesses define technical excellence.

But beneath the surface of faster delivery and automated output, there's a growing risk that's not always visible: when code is built with little oversight, no shared standards, and vague architectural direction, the result is what we call vibe architecture.

- ***It's when things technically work, but no one really knows how.***
- ***When systems grow quickly, but without a backbone.***
- ***When short-term delivery starts stacking long-term technical debt.***

This kind of architecture becomes a liability. Fragile systems that constantly need patching. Features built on unstable foundations. Products that drain time, budget, and senior talent just to stay online. Eventually, every shortcut adds up—usually at the worst possible moment.

This isn't a story about avoiding AI. Far from it. Ignoring this wave would mean falling behind. The real challenge is balancing speed with sustainability. Innovation with sound engineering. Hype with what the business actually needs to scale and grow.

That's why rethinking how we hire has become a critical part of any AI strategy. Starting fast without the right experts in place can turn into a maze of spaghetti code, apps that barely work, and millions lost to broken trust, rework, and avoidable vulnerabilities.

Methodology

This report is based on surveys conducted with software developers across platforms such as LinkedIn and Reddit. We gathered insights from professionals working in various industries, roles, and experience levels, focusing on how AI is impacting their workflows, hiring expectations, and coding practices.

In addition to survey results, we incorporated real-world observations from Inclusion Cloud's recruiting and technical evaluation processes, along with third-party sources and research to enrich the analysis.

Disclaimer

The findings and perspectives presented in this report are based on survey data, technical interviews, and external research. They should be interpreted within the context of the participants' experience and the scope of Inclusion Cloud's developer hiring expertise.

While we strive for accuracy and representativeness, these insights may not reflect the realities of all industries or regions. Inclusion Cloud is not responsible for any decisions or outcomes resulting from the use of this report.

inclusioncloud
DIGITAL ENGINEERING

About Us

We specialize in designing and delivering integrated, AI-ready software solutions that connect platforms, automate workflows, and support long-term growth. Our approach combines engineering expertise with deep knowledge of enterprise systems to deliver real business outcomes.

Through our strategic partnerships with ServiceNow, Salesforce, Oracle, and SAP, we create robust, secure, and future-ready architectures that unlock the full potential of enterprise platforms.

Discover how we're helping companies build what's next at **inclusioncloud.com**