

Measuring AI in software development: Interview with Jellyfish CEO Andrew Lau

As AI transforms coding, how will companies ensure it is actually effective? Jellyfish CEO Andrew Lau shares his thoughts on how software development workflows will need to change moving forward.



The promise of AI in coding is enormous, but how will companies need to evolve to use AI tools effectively? McKinsey Senior Partner Martin Harrysson and Partner Prakhar Dixit spoke with Andrew Lau, CEO and cofounder of Jellyfish, on how organizations are changing to achieve greater productivity through AI. Andrew shares his perspective on how the product development life cycle (PDLC) is transforming, what challenges companies face in adopting AI tools effectively, and how changing roles, processes, and measurement approaches will define the next phase of developer productivity.

This interview has been edited for length and clarity.

Prakhar Dixit: Could you tell us a bit about your and Jellyfish's growth and journey?

Andrew Lau: I am an engineer by training who caught the entrepreneurial bug about 15 years ago. After founding and working at a few companies, I cofounded Jellyfish in 2017. We're a Boston-based company of a few hundred people—about half are remote—focused on helping companies with software engineering and, more recently, managing AI transformations. Our core belief from day one was that software engineering is strategic and expensive yet often disconnected from the business. We help our customers understand which AI tools fit best for their teams and how they can measure their efficacy.

Before the COVID-19 pandemic, we were 19 people serving 20 customers—it was a very different time. The big inflection point came in Q1 2020 when everyone went remote. Suddenly, leaders were asking, "What is everyone doing?" They could not see teams, so they needed visibility and telemetry into engineering activity.

Then 2023 brought another shift. Even though it was a tough time for tech, "efficiency" and "productivity" stopped being dirty words. Everyone wanted to understand impact.

Now, with AI in software development, we are seeing another tailwind. I fundamentally believe the software development life cycle will be completely redefined within three years. That is exciting and scary for the industry.

Martin Harrysson: How do you see AI changing the field?

Andrew Lau: I do not want to sound overly dramatic, but this is one of the biggest transformations since the invention of the internet.

Over the past 20 years, software development followed a consistent pattern: setting up requirement generation and issue tracking, developing within IDEs [integrated development environments], organizing changes through source control, using CI/CD [continuous integration and continuous development] practices, testing, deploying software, and, finally, observing code post-deployment. This entire workflow could look very different in three years.

The shift is not just technical—it is human. Roles, skills, and collaboration models will all evolve. That is why I think this change is bigger than agile or cloud. It is redefining what people actually do.

Prakhar Dixit: A lot of companies are thinking about how roles and skills will evolve over time. How do you see that happening?

Andrew Lau: Look at how people work with AI. When someone writes and refines a prompt, they are effectively writing a specification by defining behavior, constraints, and outcomes. This starts to resemble architecture management or even product management. So what happens when a developer's role begins overlapping with that of a product manager or architect? What happens when a product manager's role begins to overlap with that of a designer?

For established organizations, that is a tough conversation. But newer companies are not bound by legacy structures. They can evolve roles naturally. Team ratios and responsibilities will change over time, likely through hiring cycles rather than sudden reorganizations.

Reality bears this out. Across the 600-plus organizations we track, more than 60 percent see at least a 25 percent productivity improvement from AI.¹ That is progress, but not a breakthrough.

True impact comes when adoption is deep and organization-wide. Our research with OpenAI showed that companies with 80 to 100 percent developer adoption saw gains of more than 110 percent.²

Adoption requires enablement, training, and sometimes cultural nudges. Then you hit a second wall: ensuring you have the right tools and models that fit each codebase and team. The third and hardest phase is redefining roles, processes, and incentives. Newer companies do not face that friction—older, larger ones do.

Prakhar Dixit: Assuming organizations go along this path and start shifting their operating models, how should they measure productivity and efficiency in this new environment? Is it similar or radically different from what they were doing before?

Andrew Lau: I think of measurement of impact for AI transformations in three layers.

First, adoption. If people are not using the tools, transformation will not happen. Track usage both quantitatively and qualitatively.

Second, throughput and process efficiency. Look at metrics such as pull-request rate, cycle time, and latency. These are not perfect, but they provide a systems-level view.

Third, outcomes. Ultimately, it is about whether business objectives are met, such as road map progress, defect rates, and customer impact.

¹ Gail Axelrod, "Engineering in the age of AI: What the 2025 State of Engineering Management report reveals," Jellyfish, July 8, 2025.

² Nicholas Arcolano, "Jellyfish and OpenAI team up to measure the impact of AI coding tools," Jellyfish, August 13, 2025.

The [AI] shift is not just technical—it is human. Roles, skills, and collaboration models will all evolve.

Overall, I see software productivity measurement as “systems thinking.” The PDLC is a flow. If one part accelerates—say, coding—but another lags, such as review or compliance, you have not truly improved.

Telemetry across the pipeline is essential to know where bottlenecks appear to eventually fix them. Even if coding is faster, this can shift bottlenecks elsewhere, such as requirements definition, testing, or governance. Without holistic change, gains at one point can create friction downstream.

Martin Harrysson: Where do you expect the next wave of innovation in AI tooling that will solve the bottlenecks?

Andrew Lau: So far, most progress has been in IDE-based tools and experiences for individual developers writing code. The next wave is in code review and agentic systems that coordinate work across stages of the PDLC.

Testing is a particularly interesting frontier. You cannot fully automate quality without defining “truth,” and many legacy systems lack that specification. That is a blocker for full-scale AI transformation. We will need new ways to formalize intent, essentially redefining how “correctness” is expressed.

This conversation is especially important in regulated industries, where you must prove independence and accuracy. Can an agent count as an independent validator? We are not there yet.

Until then, organizations must embed controls earlier in the process. AI does not remove the need for compliance. It heightens it.

Prakhar Dixit: Given all this, how do you see the balance between human and AI roles evolving?

Andrew Lau: For decades, we thought coding was the hard part. It turns out describing what to build is harder. Generative tools will make writing code easy, so defining the right intent, the “spec,” will be the core human task. That shift changes who adds value in the development

Find more content like this on the
McKinsey Insights App



Scan • Download • Personalize



process. It rewards clarity, systems thinking, and the ability to guide AI toward the right outcomes.

Martin Harrysson: As the AI-native PDLC scales, what are the biggest risks organizations should watch for?

Andrew Lau: Traditional risks—IP, security, compliance—still matter, but AI accelerates them. Code moves faster, so controls must be embedded earlier. You cannot rely on human sanity checks later in the pipeline.

The good news is we already have frameworks for managing risk. The challenge is connecting them directly to AI development workflows and maintaining visibility end to end.

It is an exciting and uncertain moment, but one full of opportunity for both technology and people. Many organizations have the right tools, but they do not know how to use them effectively. They need diagnostics, enablement, role redefinition, and change management to succeed.

AI does not remove the need for compliance. It heightens it.

Andrew Lau is CEO and cofounder of Jellyfish. **Martin Harrysson** is a senior partner in McKinsey's Bay Area office, and **Prakhar Dixit** is a partner in the Seattle office.

Comments and opinions expressed by the interviewees are their own and do not represent or reflect the opinions, policies, or positions of McKinsey & Company or have its endorsement.

Copyright © 2025 McKinsey & Company. All rights reserved.