



INTERVIEW

'The Human Cost of Coding will Come Down to Zero'

Hexaware CEO R Srikrishna on how AI-driven development could change the face of enterprise software, reduce tech debt and shift coding economics

By PAYAL GANGULY

V

ibe coding has transformed non-coders into developers and democratised prototyping. While it is not the perfect solution to build a business on or deploy at enterprise scale, its future holds promise.

Today, vibe coding falls short as complexity increases—whether through complex programming, building on top of existing application interfaces, or handling multiple lines of code. Using vibe coding in these scenarios reduces efficiency, as the overhead of verification, debugging, and iterations increases, along with the need to manage third-party APIs and integrations.

Tech debt, commonly used to refer to the accumulation of unmaintainable and untested code, builds up over time, from prototyping to production, which currently restricts vibe coding



largely to the prototyping stage.

However, Hexaware CEO and Executive Director R Srikrishna believes the economics of coding will change significantly in the future. He tells *Forbes India* that as AI improves, the human cost of coding could decrease significantly. Instead of rewriting or modifying existing code to solve tech debt, companies may eventually discard old code entirely and generate new code from scratch using AI.

The publicly listed company, which launched RapidX in partnership with Replit to turn natural-language descriptions into production-ready software, and its Zero License product

aimed at helping enterprises replace licensed SaaS with agentic AI software by 2026, is also developing a Zero Tech Debt product. Edited excerpts from an interview:

Q How is vibe coding evolving from a tool for prototyping to actual productisation at an enterprise level?

Initially, vibe coding started as a mechanism to prototype and stop there. But the evolution is that it goes from there [prototyping] to getting a full production-ready code. We launched a joint product with that exact objective from prototype to production. All vibe coding platforms also have agents for coding, for testing. Likewise, all code generation platforms are also going into vibe coding—we will see a convergence of capabilities.

Q How efficient is vibe coding today and will it eventually replace the need for human coders?

Enterprises typically define their criteria for coding or coding standards, including naming standards, security standards, architectural standards and others. You can feed all of this into a platform and the prompts; the platform generates code incorporating these standards. The code generated still needs work and human intervention.

Think of three dimensions—first is a hobby coder looking at a theme or programmes for an enterprise. Second is the number of lines of code associated with each program, and the third is greenfield development—which means what you are building does not interface with existing applications. As you progress on each of these dimensions with multiple themes, more lines of code and brownfield interfaces, the productivity driven by AI reduces.

At the core, a single team working

“Tech debt can come from forward engineering and reducing tech debt is one of the best outcomes of AI.”

on half a million lines of greenfield code can drive 60 percent efficiency from AI. As you go up, with say 60 teams of eight members each, working on 40 applications that are brownfield with billions of lines of code, the productivity comes down.

With our product, the promise to customers is that even at the most complex levels, you will get 30 to 40 percent efficiency, and that's the journey we are undertaking. Vibe coding is not the only answer; it is an important first step in the journey.

Q We often hear of tech debt accumulating due to vibe coding. How can that be avoided?

In the most fundamental approach, the comprehension of what is written in brownfield code is not based on LLMs. We feed that output into LLMs to get written documentation of what is in the brownfield code. LLMs by definition are not deterministic but code that is already written is. So you cannot use LLMs to discover what has been written.

The most important guard rail is that much of the code discovery is actually based on Python or other code bases, and that's what we do with RapidX. If you do that, the chances of hallucination are low. That is for reverse engineering.

Now tech debt can come from forward engineering and reducing tech debt is one of the best outcomes of AI. Today we can define the tech stack in your architecture standards and incorporate that into the markdown file and generate new code using the coding agent. If you are solving tech debt, you are rewriting

and changing some components of the code. What will happen in the future is that you throw away the code completely and generate new code based on a new tech stack. It is like recoding the whole thing again, because at some point, the human cost of coding will come down to zero. There is still AI and the token cost, but that will also move in the same direction presumably.

Q What will be the impact of all this on pricing for coding?

There will be two shifts in pricing to the client—it will depend on the volume of work and the unit price on labour and both will move in different directions.

The volume of work will reduce for a given piece of code, but the skill of the engineer will go up. Therefore, the pricing reduction will not be proportional to the volume reduction but [will be] lower than that. On the other hand, we believe that the total volume of software being built will go up—a great example is the Zero License product. A good chunk of the SaaS industry will get converted to agentic and custom code over a period of time. If I were to take a guess, it would take around three years for half of the peripheral SaaS to move towards custom code and agentic software. The workflows and reporting can and will move to agentic software, but the core will remain.

We often tell our customers—why are you renting intelligence with your data to SaaS platforms? Because ultimately, SaaS platforms consume customer data and what the customers are doing is renting their data and intelligence to the platform. There is a reason beyond cost, at a strategic scale, to take control. **F**

“Vibe coding is not the only answer... it is an important first step in the journey.”