

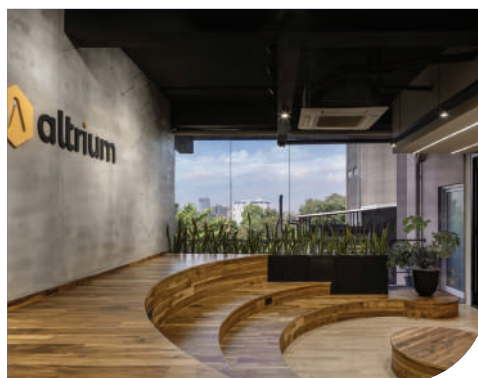
INNOVATION, SYSTEMS, AND SRI LANKA'S TECH GROWTH

INSIGHTS FROM ALTRIUM
PRINCIPAL SOFTWARE ARCHITECT
MR. THILINA ABEYRATHNA



MR. THILINA ABEYRATHNA

*Principal Software Architect
Altrium (Pvt) Ltd*



In a world where software systems quietly power almost every aspect of our daily lives, the role of a software architect goes far beyond writing code. It is about designing the invisible structures behind digital experiences and making decisions that shape scalability, performance, and long-term technological impact.

In this exclusive interview, Mr. Thilina Abeyrathna, Principal Software Architect at Altrium (Pvt) Ltd, reflects on his 14-year



journey from a curious engineering student to a technology leader responsible for shaping complex system architectures across multiple domains. Along the way, he has navigated the rapid evolution of software development, adapted to emerging technologies, and grown through both startup and enterprise environments.

Through this conversation, he offers thoughtful insights into innovation, the transformative rise of generative AI, the growing strength of Sri Lankan IT professionals on the global stage, and the mindset required for the next generation to thrive in an increasingly intelligent and interconnected digital future.



QUESTION 01: CAN YOU SHARE SOME DEFINING MOMENTS IN YOUR PERSONAL AND PROFESSIONAL JOURNEY THAT LED YOU TO BECOME A SOFTWARE ARCHITECT?

I've been in the software industry for around 14 years, and interestingly, becoming a software architect was not something I planned from the beginning.

During my school days at Pillimathalawa Central College, I was always good at academics, especially mathematics, and I really enjoyed problem solving. However, at that time, I didn't have a strong passion to become a software engineer. After my A/Ls, I entered the Faculty of Engineering, University of Peradeniya. Initially, my interest was more towards mechanical engineering. But during our first year, the Head of the Computer Engineering Department gave a talk about the software industry, its future, and global opportunities. That single

session completely changed my perspective. It made me think, "Okay, maybe software is where I belong." Even after selecting the computer engineering stream, I wouldn't say I suddenly became passionate. I followed the academic coursework, but that was all. The real turning point

came closer to my third year, when I realised that only learning academics wouldn't help me in the long run. So, I started looking for real-world exposure. That's when I got selected for Google Summer of Code, which was a very competitive global program. That experience completely transformed the way I saw software engineering. For the first time, I worked with international teams, architects, and developers. I understood how real software projects are designed, how teams collaborate, and how large systems are built. Alongside this, I also did my internship, and through these experiences, I genuinely fell in love with software engineering.

After graduating, I worked at Virtusa for about a year, and then I joined a startup. That startup environment was another major turning point in my life. With only a few engineers in the team, I had to take responsibility for everything: design, development, testing, and even client communication. That sense of ownership accelerated my learning, and over time, I grew into leadership and architectural roles.

I eventually became a Principal Software Architect after around 11 years of experience, and that journey taught me that growth happens when you take ownership, step out of comfort zones, and continuously improve.

QUESTION 02: HOW DO YOU DESCRIBE THE ROLE OF A PRINCIPAL SOFTWARE ARCHITECT?

Many people think a software architect's job is just about technology, but it is much broader than that.

The first part of my role is obviously technical. I work on system architecture, choosing the right technologies, designing scalable systems, ensuring performance and efficiency, and preventing future technical debt. Large software products don't exist for one or two years, some projects I've worked on started in 2018 and are still evolving, and others began in 2022 and will probably continue for several more years. So, making the right architectural decisions at the beginning is critical.

The second part of my role is strategic and organisational. As a Principal Software Architect, I don't just serve one team. It's a company-wide role. I help guide multiple teams, mentor engineers, define design patterns, and ensure that technology aligns with business goals.

If architecture is not planned properly at the beginning of a project, it leads to poor design, scalability problems, and a lot of technical debt. Fixing those later becomes extremely difficult and expensive. So, my job is to guide things in the right direction from the very beginning.

QUESTION 03: WHAT KEEPS YOU INSPIRED AND MOTIVATED TO GROW AND INNOVATE IN SUCH A RAPIDLY EVOLVING TECHNOLOGICAL LANDSCAPE?

What truly keeps me inspired is the fact that learning never stops in this field. Even after more than ten years in the industry, I still find myself in situations where I feel like a beginner again, and surprisingly, that is what excites me the most.

Technology, especially with the rise of generative AI, is evolving at an unbelievable speed. Things that were considered cutting-edge a year ago

already feel outdated today. If you don't continuously learn, you don't just "fall behind" slowly, you fall behind almost instantly. That constant pressure to adapt and improve actually motivates me rather than discourages me.

Another important factor is curiosity. I strongly believe curiosity is what drives innovation. When I see a new AI tool, a new architecture pattern, or a new framework, I don't just want to use it, I want to understand how it works behind the scenes. That mindset pushes me to experiment, read, test, and apply what I learn in real projects. For example, before generative AI became mainstream, artificial intelligence was mostly limited to research or specific enterprise implementations. But now, it's changing the way we design user experiences, write code, and even think about software. Seeing how fast this transformation is happening and being part of it is what keeps me motivated every day.

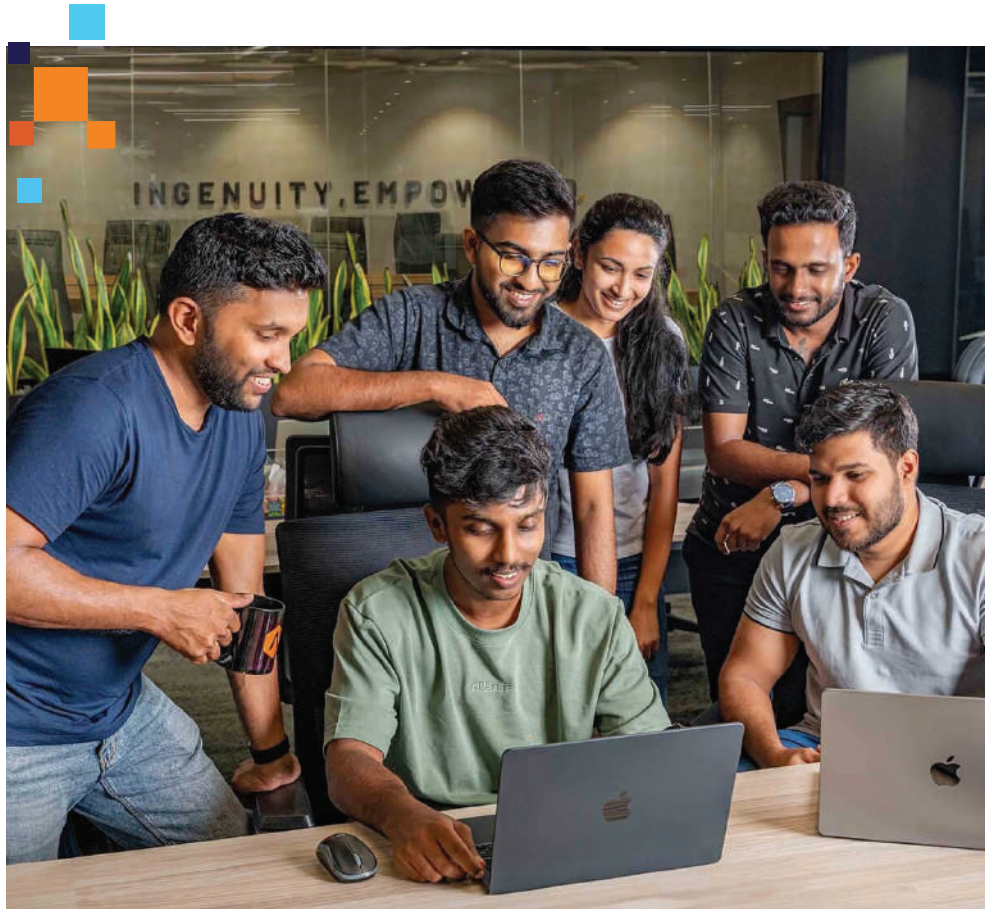
QUESTION 04: HOW IMPORTANT IS CREATIVITY IN SOFTWARE ARCHITECTURE AND HOW CAN STUDENTS DEVELOP MORE CREATIVE APPROACHES TO PROBLEM-SOLVING?

Creativity is actually one of the most underestimated qualities in software architecture, but in reality, it is absolutely essential.

When you solve real-world problems in software engineering, there is rarely only one "correct" solution. Two architects can look at the exact same problem and design two completely different systems, and both systems might work perfectly. The difference comes from how creatively they think, how they apply existing knowledge, and how they connect various ideas together.

Software architecture is not about memorising design patterns or following templates. It is about understanding those concepts deeply and then applying them creatively based on the problem, the team, the scale, and the future requirements of the system.

For students, creativity develops when



they stop trying to find the "one correct answer" and start exploring multiple possibilities. When you solve a problem, don't settle with the first working solution. Try to approach it from different angles. Try different technologies. Break things. Fix them again. This exploration is what builds both creativity and confidence.

Another important aspect is exposure. When students interact with people from the industry, they get to see how real systems are designed and how creative thinking is applied at a large scale. That can completely change how they approach technical problems.

QUESTION 05: WHAT FOUNDATIONAL SKILLS AND EXPERIENCES WOULD YOU RECOMMEND FOR STUDENTS AND YOUNG PROFESSIONALS WHO ASPIRE TO BECOME SOFTWARE ARCHITECTS?

Becoming a software architect is not something that happens overnight. It is not a position you step into right after

graduation. It comes with years of experience, mistakes, learning, and gradual growth.

Everything starts with becoming a strong software engineer. If your foundation is weak, it is very difficult to grow into an architect role later. Students should focus on truly understanding how data structures work, how systems behave under pressure, how APIs are designed, and how performance and scalability affect real systems.

What I often notice is that students are very focused on just writing code that works. But in the industry, writing code that works is not enough. You must also think about how efficient it is, how much it will cost to run, how it will scale when users increase, and how it will perform under heavy load.

With the rise of generative AI, cost has also become a very important consideration. Every AI request has a financial cost. So as a future architect or even as a software engineer, you need to think beyond technical correctness and consider system sustainability as well.

Ultimately, if students build strong

fundamentals and learn to think in terms of systems, architecture becomes a natural progression over time.

QUESTION 06: HOW DO YOU SEE GENERATIVE AI RESHAPING THE WAY SOFTWARE SYSTEMS ARE DESIGNED AND DEVELOPED?

The difference between how we developed software five or ten years ago and how we do it today is massive. Earlier, we had to remember syntax, write everything manually, and spend hours just getting a basic feature done. Now, with generative AI tools, we can generate code in minutes, sometimes even seconds. But this doesn't mean software development has become "easy." It has just changed the nature of our job.

Instead of spending most of our time writing code, we now spend more time reviewing, evaluating, and improving code generated by AI. Our role has shifted from pure creation to critical thinking. We need to understand whether the code follows good design patterns, whether it is secure, whether it is scalable, and whether it fits into the bigger system.

I sometimes describe this transformation as Software Development Version 3.0. First, we had low-level machine programming. Then we moved to high-level languages like Java and C++. Now, we are moving towards using English as a programming language, where we describe what we want, and AI generates it.

But this also increases the risk of over-dependence. If people start generating code without understanding it — what some now call "vibe coding", then they won't be able to maintain or fix systems later. In real-world production environments, that can lead to serious failures.



QUESTION 07: AS GENERATIVE AI BECOMES MORE COMMON IN EDUCATION AND WORKPLACES, WHAT SHOULD STUDENTS UNDERSTAND BEYOND BASIC USAGE?

Simply knowing how to use ChatGPT or any other AI tool is not a special skill anymore. Everyone can do that. What really matters is what you do with the output.

Students must understand that AI systems are not perfect. Sometimes they produce incorrect code. Sometimes they use outdated methods. Sometimes they generate unnecessarily complex solutions. If a student blindly accepts everything AI gives, they are not learning, they are just copying.

I've personally experienced situations where AI generated thousands of lines of code for a small task. But when I studied the framework properly, I realised the same task could be completed with a few simple lines. If I had just accepted what the AI gave me, I would have ended up with an inefficient and bloated system.

So what students really need is the ability to evaluate, question, and verify. AI should be used as a powerful assistant, not as a replacement for thinking. If you don't build your own understanding, you won't even be able to judge whether the AI is right or wrong.

QUESTION 08: AS SRI LANKA STRENGTHENS ITS POSITION AS A GLOBAL TECHNOLOGY HUB, WHAT STRENGTHS DO YOU SEE IN SRI LANKAN IT PROFESSIONALS ?

From my experience working with professionals from countries like India and the United States, I have observed some unique strengths among Sri Lankan software engineers.

One of the biggest strengths is adaptability. When new technologies or new domains are introduced, Sri Lankan professionals adapt very

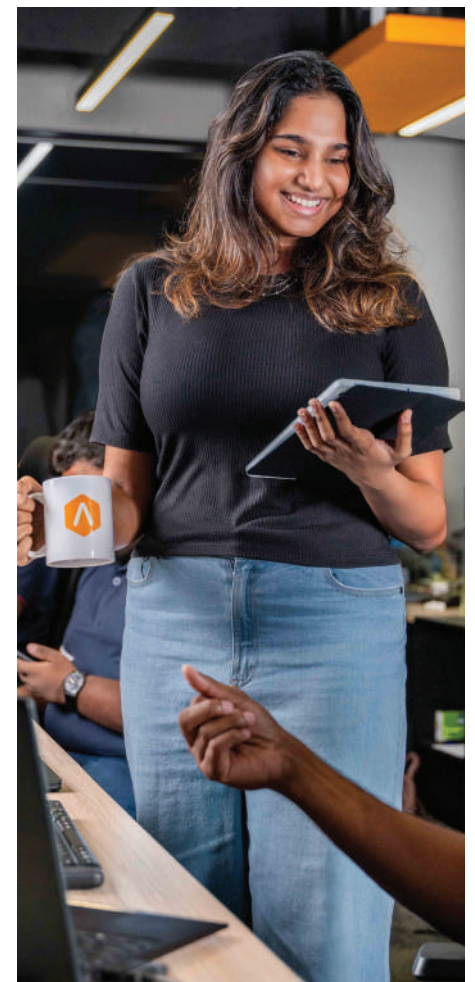
quickly. They are flexible, curious, and willing to learn.

Another strong quality is the sense of ownership and responsibility. Even when working on foreign clients' products, many Sri Lankan engineers treat the product as their own. They take pride in what they do. They don't just want to deliver; they want to deliver something good.

This mindset is not always easy to find, and I believe it makes Sri Lankan IT professionals very competitive in the global market.

QUESTION 09: IN YOUR OPINION, HOW CAN ACADEMIA AND INDUSTRY WORK MORE CLOSELY TO PREPARE STUDENTS FOR REAL-WORLD CHALLENGES?

There is naturally a gap between academia and industry, mainly because university curricula cannot change as





fast as technology does. Changing a syllabus can take many years, while the industry changes every few months. However, there are many practical ways to bridge this gap. One of the most effective methods is bringing industry professionals into universities, not just for talks, but for actual technical sessions where they show how real systems work in practice. Another important approach is involving students in real industry projects while they are still studying. Even small exposure to production-level software systems can make a huge difference in how students think and work. When students understand how what they learn in class is used in the real world, their learning becomes more meaningful and more motivated.

QUESTION 10: FINALLY, WHAT EXCITES YOU MOST ABOUT THE FUTURE OF TECHNOLOGY, AND WHAT MESSAGE WOULD YOU GIVE TO STUDENTS ENTERING THE INDUSTRY ?

What excites me most is the uncertainty and speed of change. The competition among AI companies, the release of new models almost every month, and the way technology is entering every field; from healthcare to climate forecasting, makes this a very dynamic and exciting time. It feels like we are living in an AI gold rush, similar to historical gold rushes where people rushed to mining towns. But now, instead of gold, the resource is ideas and innovation. If you have a good idea today, you don't need massive resources, you can build something powerful with the tools available. But at the same time, I always remind students not to get carried away by tools alone. Tools will change. Technologies will change. But strong foundations, clear thinking, and curiosity will always stay relevant. If students focus on building those, they won't just survive in this fast-changing industry; they will become the people who shape its future.

INTERVIEWED AND COMPILED BY:
*MS. DISNIKA RAJASOORIYA AND
MR. KRISHANTHA RANAWEEERA,
(LECTURERS AT THE FACULTY OF COMPUTING)*

LAYOUT DESIGN BY:
*CHARITH LAKPRIYA
(UNDERGRADUATE - BSC(HONS) COMPUTER SCIENCE,
FACULTY OF COMPUTING)*

