

smartShift CTO Jagdish Sahasrabudhe on Custom Code, Clean Core, and the Path to Enterprise AI



Jagdish Sahasrabudhe
CTO
smartShift

Custom code occupies an uneasy place in the SAP ecosystem. The vendor steers customers toward fit-to-standard, and clean core has become shorthand for shedding as much customization as possible. Yet for most enterprises, that code carries decades of business logic—differentiating processes, regulatory adaptations, and industry-specific rules that standard software does not deliver.

Jagdish Sahasrabudhe has spent 31 years working in that gap. After roles in solution engineering and product management at SAP, he built a startup in the partner ecosystem before

returning to SAP as a field chief technology officer. He joined smartShift in 2024 as CTO and now runs research and development for the company's automation-led custom code modernization platform.

“There is no such thing as plug-and-play in the enterprise applications world,” Sahasrabudhe said. “There will always be differences.”

ASUG spoke with Sahasrabudhe about why clean core is a continuous discipline rather than a one-time cleanup, what dual maintenance protects during a multiyear transformation, and why enterprise AI adoption hinges on custom code readiness as much as data readiness.

This interview has been edited and condensed for length and clarity.

Q: For ASUG members who may not know you, can you tell me about your journey into SAP modernization and what brought you to smartShift?

I have been in the SAP ecosystem for about 31 years now, and I started with SAP in the solution engineering and product management kind of space. All along the journey, there was always this disconnect between what a standard software vendor like SAP thought—that customers would deploy software as they have developed it—and the customers' view, which always was that “no, we will adapt it to our needs,” and that gap was not reducing; it was only increasing.

smartShift was working in exactly that area, helping customers continuously adapt their custom code to SAP's policies and evolving technology standards.

And so, the area of custom code management really interested me, which is what brought me to smartShift.

Q: What do you think SAP customers misunderstand most about clean core?

The biggest thing is that they believe that clean core would be similar to a legacy core migration journey, and part of it is because smartShift has made it so easy to do the migration. We literally do it in a matter of weeks. So, the natural question that customers ask us: “Well, if you could do that, can you just not make my system clean?”

That is the biggest misunderstanding, because clean core is a journey. It's a continuous cleanup effort. The biggest dependency is on what SAP has released as clean APIs. There are not a lot of released APIs that

map to customers' requirements. It is not a one-time, overnight, a-few-weeks, just-get-clean kind of effort. You have to put governance in place so that you are not making new unclean objects. You have to take it one transformation project at a time.

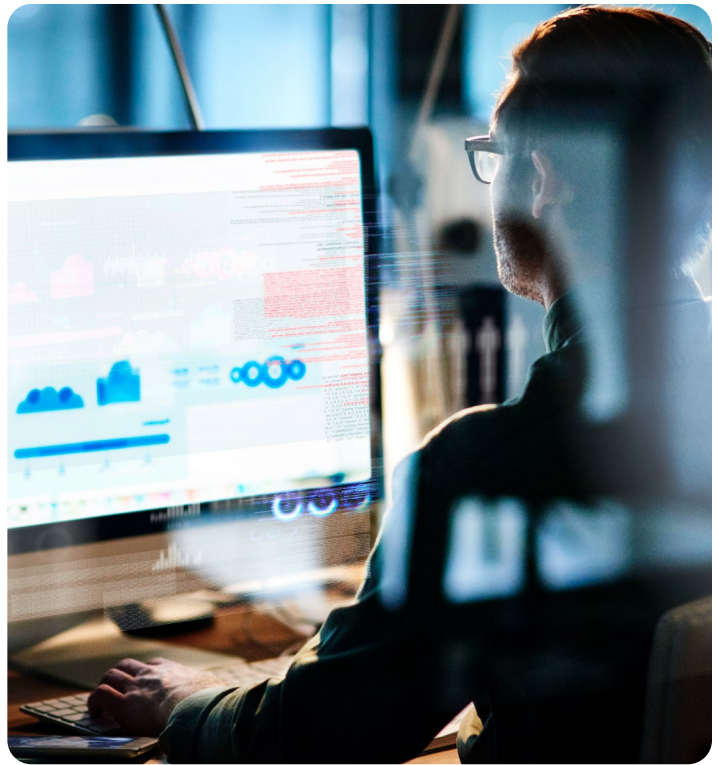
Q: A lot of the ecosystem treats custom code as technical debt, and SAP itself points customers toward fit-to-standard. Why do you see it differently?

The ideal way of adopting SAP software is with standard capability. That is the best way. Unfortunately, that is not the reality. There's a huge gap: what customers want to do to differentiate, to conform to regulatory requirements, and, honestly, their own business processes that the standard software does not bring to the table out of the box. That's why we continue to believe that we shouldn't look at custom code as the necessary evil. It has value. It has intellectual property that the customer has developed that needs to be carried forward in a very programmatic and well-managed manner.

Q: How do you separate the custom code that carries business logic worth keeping from the code that should be retired?

It all depends on usage. What we do is we basically run a variety of diagnoses on how much a customer system is using the custom code that they've created. We typically take a year-plus of data—diagnosis data, usage data—and map it to the custom code that a customer has in their repository, and that gives us a very clear picture of what can be retired safely, what is not being used.

There are various fully tested ways to safely decommission, and the reason for decommissioning is that, when you're moving to a new system, you are not carrying forward that technical debt. The first step is to diagnose it properly, and once it is confirmed that this is no longer needed, you have to safely decommission that technical debt.



Q: What are the earliest warning signs that an SAP transformation is underestimating custom code complexity?

At a very high level, looking at it and saying, well, "We have X-thousand objects; that should not be a problem," and moving ahead with that is the biggest mistake. The problems of what that custom code is doing in your system are evident only when you drill down deeper into that custom code. Is it secure? Is it opening various holes for hackers to come through the custom code into your system? Is it performant enough, or is it dragging down your system's performance?

The second biggest problem is underestimating all the priorities that SAP has highlighted. When SAP does a code analysis, it gives you, "This is priority one; you must fix it"; P2, P3, and so on. You can kick the can down the road, saying, "I know this is a problem, but let me just move on." It is going to come back to hurt you later on after the migration. Transformation projects come once every seven years, minimum, maybe 10. Fix the entirety of your custom code problems while you're doing that.

Q: These programs often run a year or more, and the business doesn't stop. What starts to break when development and production drift apart, and how do you prevent it?

The transformation projects run multiyear and the business needs continuity during that period. It's like changing the engine of your car while the car is running. Businesses will continue to keep changing their requirements, and therefore, custom code continues to be developed and updated. There is a continuous keeping-those-two-systems-in-sync activity that needs to happen.

smartShift calls it dual maintenance. While the project is going on, we will do a one-time cutoff to S/4HANA, and then, for the remainder of the transformation phase, anything that you update in the ECC system, we will auto-fit to the migrated system, so that when you do go live, you are going live on the most current set of your custom code repository. That is critical because, otherwise, everything that the customer has done during the transformation project will be lost.

Connect with Jagdish Sahasrabudhe on LinkedIn.



Q: Moving beyond the migration itself, what does modernized custom code let a customer do that they couldn't do before?

Modernized custom code is no longer a good-to-have. Non-clean code, not properly migrated code, is actually the biggest barrier to adoption of artificial intelligence. As SAP releases its AI agents, AI assistants, and any other modern functionality, it is going to expect your custom code to be in a certain shape and form. It is going to expect it to be clean. It is going to expect it to be cloud-ready, because otherwise, the AI cannot access what it doesn't know exists.

Any custom code that is hidden deep into your repository that the AI cannot even see—it won't know how to access that custom code. Additionally, these S/4HANA systems that people are migrating today are going to keep on getting upgraded in the future as well, every five years, every seven years. Getting a proper handle on custom code and making it clean enough, making it cloud-ready, is going to dramatically drop your upgrade time.

Q: Everybody's talking about data readiness for AI right now. Why is custom code readiness just as important?

Data is the absolute minimum that needs to be ready for AI to consume, but in an enterprise system, a lot happens beyond the data model. What about the logic? What about the workflow? What about the customer-specific rules, the business rules? How does AI access the logic part of it? That's where custom code comes in, because customers have indeed written an enormous number of customizations across the board, for finance, for logistics, for their industry-specific requirements. So, for all of that to be considered when AI is running, there is a gap, and your AI will not be successful unless you get a handle on your custom code.

Q: Where is AI delivering value in SAP modernization today, and where might the industry be getting ahead of itself?

AI is delivering value across the SAP ecosystem. SAP has released a variety of what it calls Joule agents, and SAP is definitely all in on the autonomous enterprise. Enterprises want 100% accuracy with their business information, because your materials management, your logistics, your finance don't run on near-enough accuracy.

The innovation of AI came from the unstructured world, from documents and from websites, so saying, well, "If it works for the unstructured world, it should work for the enterprise apps world as well" is the biggest problem today. You can already see some signs in the industry, where Anthropic and OpenAI and others are now forming partnerships with the likes of SAP and Salesforce, because they also realize that just having a model will not solve enterprise problems.

Q: You've called enterprise architecture something that's alive. For an ASUG member going live on S/4HANA next year, what should their custom code governance look like the year after that?

Enterprise architecture being "alive" means moving away from a project mindset, where custom code is reviewed every five to seven years, toward continuous enterprise hygiene. Organizations need regular visibility into their custom code's security, AI and cloud readiness, and whether technical debt is growing or shrinking. Through a live dashboard and regular assessments, they can drive ongoing transformations, remove technical debt, and make informed decisions about what to keep, redesign, or decommission. An enterprise is alive when it continuously transforms, stays fit for purpose, and remains ready for what's next.

Q: Before we wrap up, is there anything we haven't covered so far that you wanted to bring up?

When we talk about modernization and when we talk about AI, there is no such thing as one definition of AI. As soon as you talk about enterprises, back to my comment about accuracy, that leads us to a definition of AI that is split into two buckets. There is a deterministic AI component, which establishes the accuracy and the grounding. Above the deterministic layer is actually the generative AI layer, where we should let the new research in AI think on its own and make associations.

That architecture—deterministic first, ensuring accuracy, and then building generative AI on top of that—is something that the industry needs to come to terms with. You can't go purely generative, because it will hallucinate; it will not give you accuracy. And if enterprises don't trust that AI, they will never adopt that AI. On the other hand, you can't even go fully deterministic, because then that is what the older SaaS world was: 100% deterministic. So, you need a combination, and smartShift is really proud that our architecture is exactly geared toward that combination.

Visit the smartShift website.





smartShift is a global leader in SAP custom code transformation for SAP S/4HANA and Clean Core. We empower businesses to modernize their SAP landscapes, eliminate technical debt, and embrace future-ready development. Our automated and patented AI-powered solutions deliver secure, stable, and optimized code addressing all custom code issues, not just those prioritized for manual remediation, with a fixed-price, fixed-timeline, and 100% quality guarantee. Trusted by SAP's largest customers, smartShift has a proven track record of modernizing over 3,300 SAP systems and converting more than 3.5 billion lines of code. Learn more at www.smartshift.com.



ASUG is the world's largest SAP user group. Originally founded by a group of visionary SAP customers in 1991, its mission is to help people and organizations get the most value from their investment in SAP technology. ASUG currently serves thousands of businesses via companywide memberships, connecting more than 130,000 professionals with networking and educational resources to help them master new challenges. Through in-person and virtual events, on-demand digital resources, and ongoing advocacy for its membership, ASUG helps SAP customers make more possible.