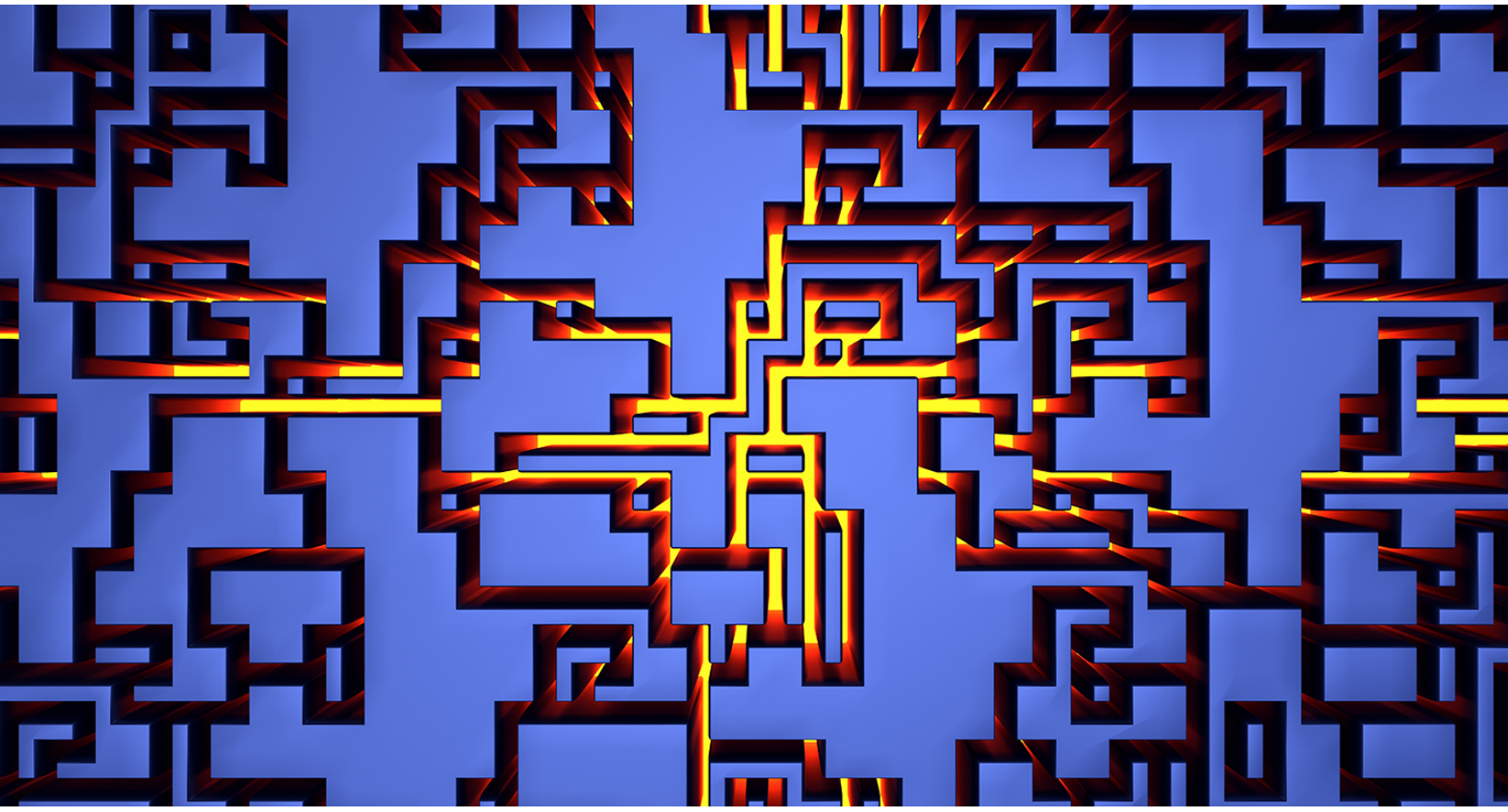
McKinsey & Company

Financial Services Practice

Pause, pivot, or accelerate: SaaS in the age of agentic AI

Agentic AI is reshaping the case for modernization in insurance. Leaders from Guidewire, Bessemer Venture Partners, and SAP discuss how insurers can respond, considering their existing commitments.



October 2026

For the past decade, when an insurance carrier said it was "modernizing," it usually meant it was transitioning from a legacy system to a modern software-as-a-service (SaaS) core. That was a sound strategic bet. But the rise of agentic AI is reshaping the assumptions that have driven these decisions—the build-versus-buy calculus, the value of a vendor road map, and the way software is priced.

Boards are now asking, is it better to accelerate or pause these commitments, or is it better to pivot toward another play entirely? To work through these questions, we convened a panel of three industry experts, in which McKinsey Partner Sanjay Kaniyar talked with John Mullen, president at Guidewire; Caty Rea, vice president at Bessemer Venture Partners; and Lance Senoyuit, senior principal, industry executive advisor, financial services at SAP.

The following conversation has been edited for clarity and length.

Has the finish line for modernization moved?

Sanjay Kaniyar: Upgrading to a cloud-native SaaS core has been the typical approach for insurance modernization over the past ten years. But given where AI and agentic are heading, is that still the right definition of modernization, or has the finish line moved?

John Mullen: Carriers should absolutely be building toward their agentic future—but in parallel, they still need to upgrade their core systems. AI helps with the speed and cost of that effort, but the effort itself remains. What becomes far more important is the destination you're designing for. The choices you make should help ensure that you land on an open, trusted core and the right architecture rather than focusing solely on the features and functions of a straight system or workflow replacement. The ability to build your own agents and have them integrate with that modern core is also important. The earlier carriers think about that architectural destination, the better they set these programs up for success.

Caty Rea: I agree with all of that. Moving to a modern core and getting onto the cloud are still necessary. Agents need application programming interfaces (APIs) and access to structured and unstructured data, but that's no longer sufficient. The opportunity is larger now. There are whole lines of work that AI agents can do. So first, you get onto the modern core and its open architecture, and then you build on top of that.

> 'Moving to a modern core and getting onto the cloud are still necessary . . . but that's no longer sufficient.'
>
> —Caty Rea

What still holds, and what you are really buying?

Sanjay Kaniyar: Which of the original reasons for going to a SaaS core—configurability, cloud, faster product launches, lower maintenance—still hold in an agentic world? And is what you buy as "SaaS" becoming something fundamentally different?

John Mullen: For a long time, the decision to replace a core system was usually made under duress—the threat to the longevity of the existing system and questions about maintainability and security. All of those problem statements still hold, but the pressure has shifted. For decades, it was the operators telling the board about the demise of aging systems. Now it's top-down: We have to go faster to compete. Take all those old reasons and multiply them by the speed of competition—particularly in personal lines, where pricing agility and product speed to market are the name of the game. If you don't solve these challenges while also building your core, you'll accumulate real tech debt.

Caty Rea: And the payoffs can be larger now. The old ROI equation, a multiyear lift-and-shift, could be hard to justify. Now you can move to a modern core and then capture incremental benefits as the AI market evolves. And one historical challenge—the mapping of various workflows, implementations, and migrations—is starting to ease. AI is compressing the cost of the migration itself, data mapping, test generation, and configuration, so some of the old downsides of moving off legacy systems should reduce over time.

The new economics: Costs shift; they don't fall

Sanjay Kaniyar: Modernization has always been as much an economic decision as a technological one; license fees, implementation costs, and total cost of ownership all come into play. How does agentic AI change that equation? And how is the economics of SaaS itself shifting from per-seat and per-module pricing toward consumption- or outcome-based models?

Lance Senoyuit: Costs shift; they don't really fall. With agentic AI, implementation genuinely gets cheaper. Migration, test generation, and field mapping are exactly what this technology is good at. So if you are scoping today using systems integrator rate cards or pricing benchmarks, you are likely in the wrong ballpark and on the high side. What doesn't fall is running cost. It changes shape as seats give way to compute, tokens, orchestration, and a governance function most carriers haven't budgeted for. This means we all have to be much better at our business cases, justifying tokens by showing decreases in days per claim and exception rates, as well as software savings. We've all heard of the team that burned through roughly 100 percent of its annual AI budget in a couple of months. Your CFO shouldn't come to you after go-live and say, "I understood software costs went down, but now there's a new line item for tokens, and I had to hire four more people to govern AI."

John Mullen: Lance makes a point we should all pay attention to: If we're only going after incremental process improvements, cost-shifting persists too long and returns diminish. The bigger prize is reimagining the interaction models offered to producers and customers. The claim adjuster's experience has always been about workflow and fields. Over time, workflow moves to the background while decisions and context come to the foreground. This means we need to redesign processes to be decision- and information-based rather than workflow- and field-based if we want to reach the next level of benefit.

Caty Rea: I'd add here that even if absolute spend shifts rather than falls, across a carrier's total tech package—from the core through AI, built or procured—you should see your loss and expense ratios improve and the value delivered per dollar of total cost of ownership go up. When we look at companies to invest in, we want the pricing equation for carriers to be as simple as possible—and, especially for AI applications, increasingly outcomes-based rather than per seat. When you shift to outcomes, you say you can do the work and then prove it by delivering the outcome.

Drawing the line: Inside the core versus outside

Sanjay Kaniyar: One common theme in what you are all saying is that the business case definition will become tighter. There will be more discipline and more scrutiny. Given this, where should the line fall between what lives inside the SaaS platform and what lives outside it—in an AI orchestration layer or the carrier's own environment?

Lance Senoyuit: It's hybrid. My rule of thumb is that if in any way you'll have to defend it to a regulator, an auditor, or a court, it should live inside the system of record: statutory reporting, the subledger, and so on. The fast-moving work, including new products, agent portal customization, and customer experience, can live outside, but with thorough governance around the APIs. One caution: Be careful modifying the core. We're starting to treat every customization as a tax on AI. What you customize is what your agent can't see, so you have to be careful. From a semantic point of view, language and definitions have to be consistent across claims, policy, and finance. If you change those tables and drop in agents, you'll find very quickly that they can't perform the way you were promised.

John Mullen: The horizon will differ by carrier size, but in insurance, you cannot easily outsource, automate, or treat the core as a generic commodity; it holds far more strategic weight, control, and value than in most other industries. It defines how the product is presented and delivered to the customer or market, and it has to be explainable to regulators every day. As such, a large tier-one carrier may build differently and pay to maintain contextual continuity between the edge of the enterprise and the core. The only time this gets oversimplified is in discussions of "deterministic versus probabilistic" decision-making. Rule-based deterministic decisions are fine, but, to Lance's point, probabilistic decisions still need to be explainable, repeatable, and consistent across all economic classes, which can be very challenging.

‘The bigger prize is reimagining the interaction models offered to producers and customers.’

—John Mullen

Caty Rea: I’d add a caveat. We talk about AI generally, but we usually mean generative AI—and not everything has to be generative. For deterministic workflows, business rules still work, machine learning operations and data science can work really well, and we don’t need to reinvent the wheel. For me, the most important thing when considering where you draw the line is to architect your systems to be open and API-accessible, so if that line moves over time, you’re not stuck in a closed ecosystem.

John Mullen: I agree. Don’t sleep on traditional machine learning; it must become a foundational capability across the company’s entire product and operations portfolio if you want to scale efficiently.

Build, buy, or ‘build with’?

Sanjay Kaniyar: If AI makes building software dramatically faster and cheaper—software development being the so-called killer application for generative AI—does that flip the build-versus-buy math that pushed carriers toward SaaS? Should carriers be building more now?

John Mullen: The build-versus-buy conversation is shifting to a “build with” conversation. That puts the burden on the software provider to commit to two things: first, that we’ll forever be an open, trusted core and won’t throttle the model context protocol (MCP) and API surface area, and second, that there will be no one-way doors. We protect the integrity of the core, ensuring that updatability remains safe so whatever agents a carrier builds on top of us keep working as we evolve underneath it. Every carrier I talk to should be building more. But they also need to invest where they can achieve the greatest brand, producer, and customer differentiation while vendor capabilities advance at the same or an accelerated rate. In this context, strategic alignment with the road map is more important than ever.

Caty Rea: It's still very hard to vibe code your policy admin system or your enterprise resource planning. Those have been built over many years and have real moats. And it's not just about writing code; the bottlenecks are evaluation, maintenance, compliance, and auditability. So build and buy is 100 percent correct. Keep the core and build the differentiation layer on top that is specific to you. And allow your people to learn to build the repeatable, low-hanging-fruit workflows themselves. Build that muscle.

Lance Senoyuit: Absolutely. Building was never the expensive part. Owning it is. Time and again, I've seen a customer with three or four people who built a great agent portal, and then one of them retires. Now you've got a documentation and testing gap, and you're carrying dead weight. The key is the back end. You need to be sure that you own the software around the building and that it is not tied to one person. This means you have to think through the rationale and the full life cycle before you build.

Orchestrating a sprawl of agents

Sanjay Kaniyar: Clearly there are going to be lots of agents around. Carriers will build agents, buy agents from third parties, and get agents shipped out of the box by their SaaS platforms. If I'm a carrier getting agents from everywhere, how do I stay sane and keep my footprint manageable?

Lance Senoyuit: My advice is to think about it in three parts: the human, the AI assistant, and the agent. The human designs, dreams, and develops; the assistant orchestrates; and the agents execute. Treat a new agent like a new employee: On day one, it knows nothing, so you train it on context, give it scope, make it revocable, and keep an audit log and lineage so that ten years later, it can say which decision it made and why. And because you're running agents from third parties, vendors, and your own team, the assistant has to sit on top. Manage each one like an employee: Grow it, give it feedback, and at times, retire it.

John Mullen: Orchestration of agents is where the platform piece comes into play for providers. Two elements matter here. The first one is developing agent-builder capabilities, not to force customers to use the partner's solution but to set guardrails so any agent can both contribute to and consume context from the core. In insurance, context is king and is key to making agents work effectively, and the vast majority of that context sits inside the core. The second is to focus on how things get built. Over the last year, we built a development harness that brings the developer experience to customers, and early returns show it's about 40 to 50 percent more efficient in token consumption and throughput than going straight to the large language models.

> **'Building was never the expensive part. Owning it is.'**
>
> —Lance Senoyuit

Caty Rea: I agree that the orchestration layer is critical because things are evolving so fast, be it models, your own builds, core-vendor add-ons, or third-party tools. The orchestration layer pulls these elements together—and thanks to MCPs, that's getting easier. You actually derive more value from your core systems. But there are two principles to keep in mind here: First, anywhere there's regulation, there's more risk than benefit in reinventing the wheel, and second, anywhere you can buy something where you benefit from others' data—like a shared fraud network—the sum of the parts gets better. That's where one plus one can equal three.

What to do if you are in the middle of a migration

Sanjay Kaniyar: Most of our clients are in the middle of a migration of some kind. If you're midway through a core or SaaS modernization today, how should you think about your next steps?

John Mullen: As we've discussed, the finish line has moved, but that modern, trusted core with no one-way doors is still required. What's changed is the prize: We're no longer just talking about IT operational excellence but also about rethinking labor, processes, and services to the industry. The prize at the end is far greater than it was before, and the earlier you start redefining what you are aiming for and your pathway to achieving it, the better.

Lance Senoyuit: I'd agree with John. What I'd add is about the later stages of a project. Many programs spend remaining time and budget on creating screen parity because that's what users ask for. I'd spend it on the data model and API coverage instead; the screens have the shortest remaining life of anything in the program. Ask for API parity now, midprogram, while you still have leverage because you won't get it in year four. And in insurance, "pause and reevaluate" is rarely a pause. If the architecture is genuinely wrong, stop honestly. If it's right, finish it, but change what you build the rest of the way.

Caty Rea: Don't pause, waiting for the AI picture to settle—it won't, and what agents need sits on the other side of the modern core. They can't do their job without the clean data and connections that a modern core platform provides. But replacing your core system isn't the finish line anymore. The real payoff comes from the work that AI transforms once that new foundation is in place.

> **'In insurance, context is king . . . and the vast majority of that context sits inside the core.'**
>
> —John Mullen

Find more content like this on the
McKinsey Insights App
Scan • Download • Personalize

Caty Rea is a vice president at Bessemer Venture Partners. **John Mullen** is president at Guidewire. **Lance Senoyuit** is senior principal, industry executive advisor, financial services at SAP. **Sanjay Kaniyar** is a partner in McKinsey's Boston office, where **Tanguy Catlin** is a senior partner and director of the McKinsey Global Institute; and **Nakul Juneja** is a consultant in the Toronto office.

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

Copyright © 2026 McKinsey & Company. All rights reserved.