Who Owns the GTM System?

Everyone owns a piece of AI. No one owns the system AI runs on

By **Jonathan Moss**
Executive Vice President, Experity

WHAT YOU'LL LEARN IN THIS ARTICLE

- GTM is a system, not a set of functions.
- AI amplifies whatever already exists, both the good and the bad.
- There is a right sequence for deploying AI. Most companies are doing it backward.
- Revenue Operations is shifting from support to system ownership.
- GTM ownership is a CEO decision.

Every CEO has an AI strategy. Almost none have answered a key question: who actually owns the system it creates? Not who picks the vendor. Not who runs the pilot. Who owns the architecture: the data, workflows, agent layer, and feedback loops that increasingly determine whether your go-to-market actually works.

When GTM has gained new tools in the past, RevOps absorbed that responsibility, layer by layer. CRM administration, data hygiene, reporting, workflow management. Each element of complexity created a new process, and RevOps expanded to take it on. This is the pattern, not the exception. RevOps grows because go-to-market complexity rises.

However, this time is different. AI is the largest complexity jump yet, but it is not just increasing the workload—it is automating the work itself.

Which means the discipline now faces a fork in the road. AI will either absorb RevOps, automating away reporting, process documentation, and tool administration, or elevate it to the chief architect of the entire go-to-market system.

There is no middle path. The only version of the role that survives is one that fundamentally transforms.

The System No Longer Runs on People

Most executive teams have not fully internalized what their go-to-market has become. It is no longer a people-and-process operation. It is a system. And it is becoming a system that humans can no longer operate by hand. A modern go-to-market motion requires real-time signal detection

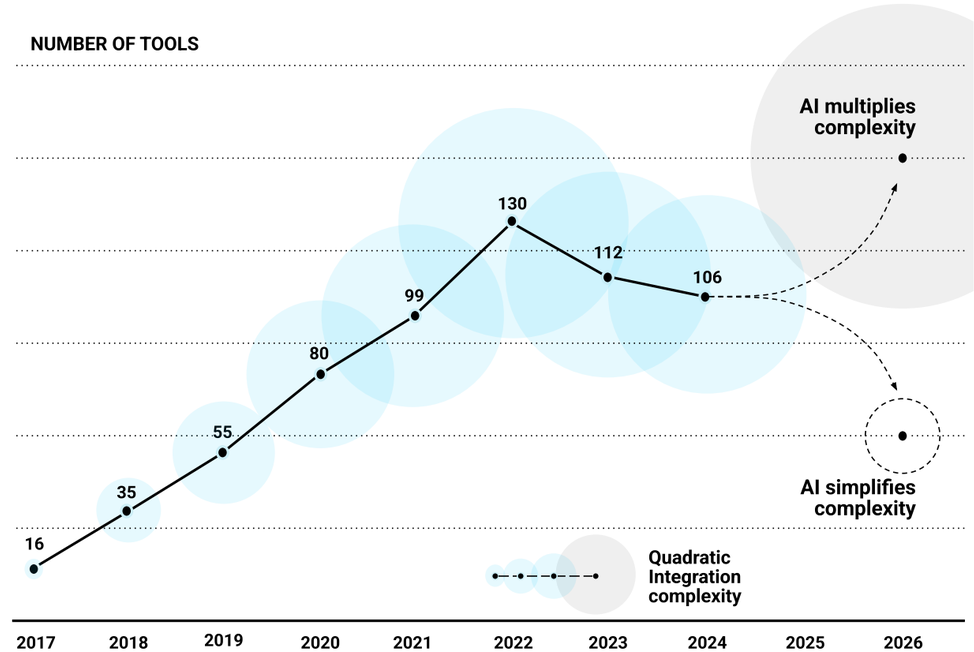


Figure 1: Tools grew linearly. Integration complexity grew quadratically. The 2026 fork shows the two paths AI creates, simplification or multiplication.

across hundreds of accounts. Dynamic lead scoring that adapts to behavioral patterns. Automated workflow routing based on segment, intent, lifecycle stage, and deal velocity. Cross-functional handoffs that need to happen in hours, not days. Customer health models that synthesize product usage, support tickets, NPS data, and billing patterns into a single score that triggers the right action at the right time.

The problem is not just the number of tools. It is the number of potential connections between them. The average company runs 106 SaaS applications as of 2024, and in a stack that size, the number of possible pairwise integration points grows quadratically. Add one tool, and you do not add one unit of complexity. You add 106 potential interconnections. The linear management systems most companies employ simply can't keep up, and the go-to-market system breaks down.

With the advent of AI, many executives believe they are on a path to consolidation. What we experience in practice suggests otherwise. Two things are happening simultaneously at most companies. RevOps is using AI to consolidate in one department, while another department adopts eight new AI tools without telling anyone. The result is complexity that quietly grows at the interaction layer as AI tools are added outside any governance structure. No single team has visibility. No dashboard tracks the whole.

The truth is no human team, regardless of talent, can manage that interaction layer manually and keep pace with it. Most signals get missed. Most handoffs happen late. Most health scores trigger action only after the moment has passed. The GTM machine of the future needs AI for external productivity, but it also needs AI for this internal management.

But while humans can't move fast enough without AI, AI can't steer the ship on its own. A person still needs to architect and orchestrate the system. A person still needs to be the translation layer between business objectives and machine execution. Why? Because AI doesn't necessarily give you the right thing; it gives you more of what you already have.

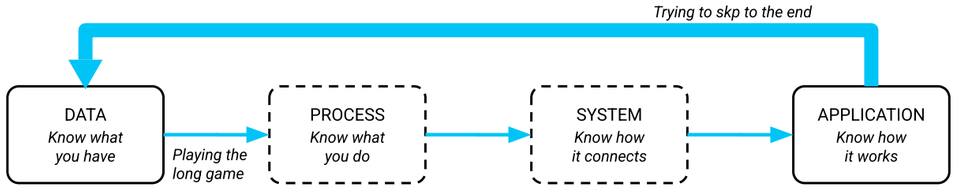


Figure 2: The sequence AI requires to deliver its best work

AI Does Not Fix What Is Broken. It Scales It.

AI is a multiplier, not a corrector. It amplifies whatever it touches. Clean processes, agreed-upon definitions, and healthy data become dramatically faster and more effective. Broken processes, inconsistent definitions, and messy data become dramatically worse, at scale.

An AI layer dropped onto a broken foundation produces outputs that look authoritative and say nothing accurate. The AI is not malfunctioning. It is doing exactly what it was designed to do: synthesizing the inputs it was given. When those inputs are garbage, the outputs are confident garbage.

I watch companies make this mistake over and over. They skip straight to the application because it has a compelling demo and a visible ROI story. What they are actually doing is starting at the end of a sequence that has to be earned from the beginning.

Four stages have to be in place to enable AI to deliver its best work: data, process, system, and application. The companies playing the long game move through that sequence deliberately. That doesn't mean each stage has to be perfect before connecting to the next, but it does mean they do them in the right order. Then they get each stage connected quickly enough that the system can start teaching them what needs to improve. Learning happens across connections, not in any single stage.

The companies that try to skip to the end do the opposite. They jump straight to the application, spot fix problems when it fails, and wonder why the gains never compound. A disconnected system cannot learn. A system built finish-to-start will always need fixing start-to-finish. Playing the long game means building in the right order to identify and scale what works. Skipping to the end means foregoing system design and blindly counting on AI to figure out what it needs to scale. It usually picks the wrong thing.

The Role That Has to Exist

To steer that process and lead the next generation of RevOps requires a new type of leader. That shouldn't be a surprise. Every major platform shift creates a new executive role. Not an upgraded version of what existed before. Something genuinely new. The CTO emerged when technology became a competitive differentiator. The CMO emerged when marketing became a system. The CRO emerged when go-to-market became too complex for sales leadership alone. Each time, a function that had been treated as operational suddenly became strategic. The market created a new seat at the table to reflect that.

We have started calling it the VP of Growth. Not a rebranded Head of Demand Generation. Not a marketing leader with a new title. A dedicated revenue operations role, someone who owns the growth model, the data architecture, the system, and the AI layer that connects functional teams around shared outcomes. This is the role Revenue Operations must grow into.

The market is already pricing it accordingly. LinkedIn shows 6,000 openings as of April 2026, with a base compensation range of $250,000–$350,000 and total compensation packages up to $550,000. That is not a coincidence. That is a market recognizing a function for the first time at its actual strategic value.

Here is the provocative part. The Chief Customer Officer was created to unify customer-facing functions under one executive. But that unification was organizational, not architectural. It connected reporting lines, not systems. The VP of Growth does what the CCO was meant to do, but with actual system authority. A new executive has entered the room.

Go-to-Market as a Product

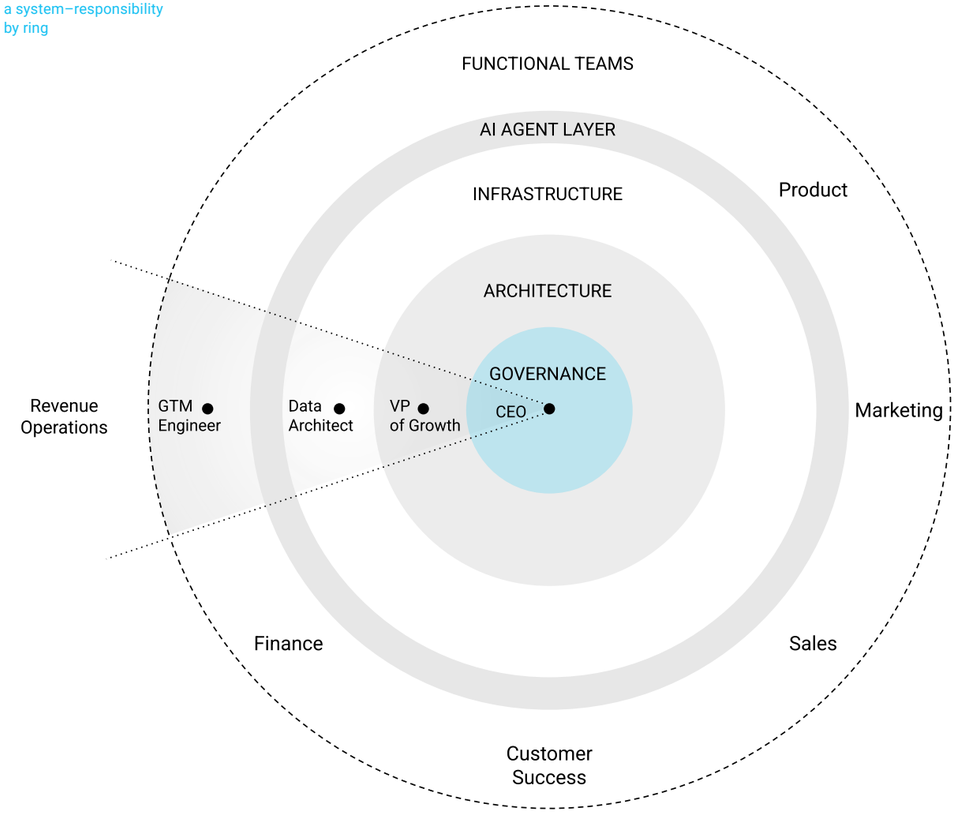
Once go-to-market is understood as a system, the organizational implications follow directly. A system needs a product organization, not a support function.

The best RevOps teams are already moving in this direction. Not by design. By necessity. They form cross-functional pods: a data engineer, a workflow automation specialist, and a go-to-market operator with deep domain knowledge. They run sprint cycles. They maintain backlogs. They do release management. They have arrived at product development vocabulary because the work demands it.

The shift is to formalize what is already emerging. RevOps stops being a centralized service desk that takes tickets from Sales and Marketing. It becomes an embedded systems organization that builds and maintains the go-to-market architecture. The same transition product engineering made, from building what the business asks for to owning the product, is the transition RevOps is now positioned to make with the GTM machine.

The CEO sits at the center of Figure 3, not because they operate the system but because they make the ownership decisions that give everyone else the authority to do their job. The VP of Growth owns the architecture. The Data Architect and GTM Engineer own the infrastructure. The functional teams operate in the outer ring, but they are not subordinate to Revenue Operations. They are the customers the system is built to serve.

Figure 3: Revenue Operations as a system–responsibility by ring



The AI Agent Layer is the GTM product interface. When the functional teams need something from the GTM machine, the AI provides it to them. It sits at the edge of the GTM infrastructure, serving as the translation layer that protects the functional teams from the underlying complexity of the stack and frees the RevOps team from having to handle manual requests for every GTM need. Just like the product itself protects the engineering team from having to manually fulfill every customer request for an output or a report.

The Capability Gap Is Real

This transition will not happen by reorganizing an org chart. It requires a genuine capability upgrade and an honest assessment of who can make the leap. The divide is real, if not uniform. Some RevOps professionals are already trending toward systems thinking and AI fluency. They are building automations, experimenting with AI tooling, and thinking in terms of architecture rather than administration. They will evolve into the systems architects this moment demands. Others are deeply skilled at the current jobs, reporting, tool management, and process documentation, but have not yet built the capabilities required for what comes next. The gap is not about intelligence or work ethic. It is about capability.

Taking the next step has a talent requirement that most companies have not yet considered. Meeting that bar will require three actions from the CEO. First, honestly assess which RevOps people are trending toward the system architect role, and what investment accelerates that trajectory. Second, build specific capabilities: data architecture, workflow design, and AI literacy (not generic AI awareness training). Third, accept that some of this capability will have to come from outside the company and cannot be developed from within.

The GTM Ownership Decision

Everything in this argument leads back to the question it opened with: "Who owns the GTM system?" And in most companies, the honest answer remains: nobody owns it. Pieces are owned by Marketing. Pieces by Sales. Pieces by IT. Pieces by nobody. The system as a whole is an orphan. You can feel it in the broken handoffs, the conflicting metrics, the tools that don't talk to each other, the AI pilot that worked in the demo but failed in production.

RevOps, led by the VP of Growth, is poised to take up that mantle, but they can't do it alone. They need to be empowered by the CEO to design your GTM the same way the engineering team designs your product. They need to be enabled to reorganize the system so that AI investments compound into structural advantage rather than scattering into point solutions that no one maintains. RevOps can be your GTM champions, but they're waiting on you, the CEO, to make the leap. Otherwise they may diagnose the problem but can never fix the structure that produces it.

Jonathan Moss is an executive leader in go-to-market strategy and revenue operations, currently serving as EVP of Growth and Operations at Experity. He specializes in building scalable growth systems through data, AI, and systems architecture, and is known for frameworks such as the "Revenue Nervous System." He holds an MBA from Harvard Business School.

Experity is a U.S.-based healthcare technology company delivering integrated software and services for urgent care and on-demand healthcare providers. Its platform combines electronic medical records, practice management, patient engagement, billing, and analytics, enabling clinics to streamline operations, enhance patient experiences, and drive stronger clinical and financial outcomes.

NOT SO TINY THOUGHT

Your system is telling you how it wants to grow. But you need to be willing to listen.

GROWTH

For more insightful articles,
subscribe today.

winningbydesign.com/growth-journal