www.pipelinepub.com
Volume 22, Issue 10

The AI Orphan Problem: Why Operational Agility Requires More Than Experimentation

By: Frank Ploumen

Telecom has spent billions on AI and has little to show for it — not because the tools don't work, but because the tools don't talk to each other. Most service providers have run pilots, stood up proof-of-concept tools, or deployed a point solution bot or automation. What they haven't done is build AI into the fabric of how they run their business.

The pressure to move is real and understandable. But what's often happening now is that urgency is driving decision-making without a long-term framework to support it. The result is a lot of activity and very little operational transformation.

Understanding why — and what to do about it — is the difference between building a genuinely competitive business and accumulating what I call "AI orphans," or isolated AI investments that work in a narrow context but can't scale, can't integrate, and can't sustain the outcomes they were designed to create.

Most AI Deployments Are Trapped Inside Silos

The reason most AI isn't truly operationalized isn't a strategy or intent problem. It's an architecture problem.

Consider what this looks like in practice. A decade ago, everyone rushed to connect devices to the internet. Lights, thermostats, doorbells, smoke sensors, each with its own app, its own interface, its own logic. The result wasn't a smart home. It was a connected home with the user in the middle, managing 27 apps that couldn't coordinate with each other. The platform integrations that eventually made smart homes actually smart took longer to build, but they're what survived. The point solutions got ripped out and replaced, and anyone who built a business around them was left scrambling.

AI is following the same trajectory, only faster.

Teams across telecom organizations — network operations, customer service, finance, marketing — are each experimenting with their own AI tools. Each team builds something that works within its own

vertical, creating the illusion of progress. A chatbot that deflects password resets. A churn model that flags at-risk subscribers. A campaign tool that improves acquisition targeting. Each solution generates real results but doesn't integrate or connect with the others.

The output of one system becomes a report, a graph, or a PDF — something that a human carries across organizational boundaries and presents in a meeting, where other humans debate it and eventually make a decision. The AI stopped working the moment it reached the edge of its silo. Despite all the automation, there's still a rate-limiting step: a PowerPoint deck or a conference room.

This is the AI orphan problem in practice. Not a single dramatic failure, but a slow accumulation of point solutions that each solved a narrow problem and foreclosed on the possibility of solving a bigger one. If you deploy assets in your operational network that are one-offs, they become very hard to maintain. You end up slowing down your overall business in the long run because you have so much burden and so many touch points.

Real Operationalization Starts With Outcomes, Not Tools

The antidote to the AI orphan problem isn't slowing down. It's reorienting around outcomes before reaching for tools.

In telecom, there are essentially four outcome categories that matter for service providers: acquiring new customers, maximizing revenue from existing subscribers, reducing churn, and optimizing the cost to deliver service. Every AI use case in the industry maps to one of these four buckets. So the question isn't which tools are available to address issues in each bucket, it is which outcomes are most urgent, and what architecture is required to serve them at scale.

Starting with outcomes rather than tools changes the questions you ask before you build. Instead of "can we deploy a chatbot quickly?" the question becomes "what does the end-to-end support experience need to look like, and what systems does any support tool need to integrate with to actually resolve problems?"

That second question is harder to answer. It requires conversations across departments and demands an honest audit of which legacy systems can participate in an integrated workflow and which ones will become the weakest link in the chain. While it takes longer to get right, it is the only question worth asking if the goal is operationalization rather than optics.

What This Looks Like in Practice

Consider the difference between a point solution and a truly operationalized AI workflow in the context of subscriber upsell.

In a point solution model, a marketing team builds a propensity model that identifies subscribers likely to benefit from a faster tier. The model generates a list. That list gets exported, reviewed, and fed into a campaign tool. A customer service rep surfaces the offer in a call. The subscriber says yes. The rep opens a billing system and manually processes the upgrade. Three or four handoffs, each introducing friction, delay, and potential for abandonment.

In an operationalized model, a care agent identifies an unmet need in real time. That signal flows to a marketing agent that generates a personalized offer. A billing agent executes the transaction. The subscriber clicks a button, and the upgrade is live.

There's nothing more frustrating than a provider that identifies an unmet need, and as soon as you

choose the solution to meet it, you're told to call the provider to complete it. And sometimes in these cases, the subscriber doesn't call, they leave, or they never convert. That one missed moment, multiplied across thousands of interactions, is lost revenue, again and again and again.

Making this work requires orchestration across systems that were never designed to talk to each other. Workforce management, billing, ticketing, provisioning — each needs to hand off context seamlessly rather than forcing a human to carry it across the gap. The good news is that the technology to do this now exists. Agent-to-agent communication protocols like A2A and Model Context Protocol (MCP) make it possible for systems to exchange context without requiring everything to be funneled into a single data lake. Two years ago, this architecture was theoretical. Today, it's becoming standard.

That said, orchestration without guardrails is just faster risk. As agentic workflows become more capable, governance becomes essential: permission-based rules, predictable behavior, clear escalation paths, and accountability for what the system is allowed to do. A billing agent that can execute a transaction needs to know when it can't; when a fraud flag should stop the process, not just slow it down. Without those controls, automation can create risk faster than it creates value.

This is why platform thinking matters so much. A platform gives the organization a consistent operating model, while point solutions often skip the structure needed for long-term reliability. In a fast-moving environment, leaders should resist the temptation to measure success by how quickly a prototype is built. The better metric is whether the workflow can survive scale, change, and integration demands over time.

The Payoff: What Operational Agility Actually Delivers

The business case for getting this right is straightforward. Subscriber lifetime value is high when loyalty is maintained. Churn is expensive — not just in lost revenue but in the cost of re-acquisition. Call center deflection, automated provisioning, and intelligent troubleshooting all reduce the cost to serve. And the ability to close an upsell in a single interaction rather than across multiple touchpoints means capturing revenue that currently evaporates in friction.

But the more significant payoff is competitive positioning. The experience gap between service providers and the tech-native companies entering the space is growing — and it's widening fast. When a satellite provider can complete a transaction on a single web page with one click, and a legacy provider requires a phone call, that subscriber doesn't just feel frustrated; they start looking at alternatives. Speeds and feeds no longer differentiate. Experience does. And once a customer decides the experience isn't worth it, no amount of product quality wins them back.

Service providers who operationalize AI — who build for orchestration with guardrails and controls rather than optimizing within silos — will be able to deliver the frictionless, click-to-buy experience that consumers increasingly expect. Those who accumulate AI orphans will find themselves maintaining an ever-growing portfolio of point solutions that can't evolve, can't integrate, and can't compete.

The right move isn't to stop experimenting. Experimentation is how assumptions get validated. The move is to experiment with the long game in mind — to ask, before anything gets deployed, not just "does this solve the problem?" but "where does this go in two years, and does my architecture support it getting there?"

The service providers who capture the AI opportunity will be the ones who treated it as an operational transformation, not a collection of tools. Think big first. Design for the outcome, not the win.