RESEARCH

The Day-Two Problem

The day you launch AI is not the hardest day. It's the easiest one.

By **Alan Babbitt** · Intelligent Systems Architect · August 2026 · 6-minute read

The premise

Launch day is the easy day.

Everyone knows this about hiring. Nobody thinks the hard part of a new employee is the offer letter. But somehow, with AI, the whole conversation stops at go-live. The pitch deck talks about pilots. The board update talks about deployment milestones. The vendor demo talks about time-to-first-value. Every one of them implies the same thing: the hard work ends when the system turns on.

That's the trap. Standing something up has genuinely gotten easy. What has *not* gotten easier is everything that happens after.

The hard day comes about ninety days later. The tool that impressed everyone starts giving different answers to the same question. The instructions that encoded one of your rules got quietly edited in a vendor update. Somebody asks who authorized the thing that just touched customer data, and the honest answer is that nobody knows. And the time you were supposed to save has been quietly eaten by the new work of cleaning up after the system.

That's Day Two. It's where most AI deployments break, and almost nobody plans for it.

> *Day One is a project. Day Two is an operating discipline. The firms that get this right plan for both.*

The three failures we see most

Three patterns repeat across organizations that launched AI confidently and then watched it quietly erode. Each one is silent for a while, then expensive all at once.

One: the drift nobody noticed. The first month is great. Accuracy is high, people are happy, the dashboard is green. Then, around month three, the answers start to change. Not dramatically. Just enough that a couple of clients push back, a few people quietly stop using the tool, and the odd awkward case gets routed around instead of through. What happened? The vendor updated the underlying model. A source document went stale. Someone edited an instruction with good

intentions and no record. None of that sets off an alarm. The system is still running. It's just no longer doing what it used to do. By the time anyone in charge notices, six months have gone by and trust in the tool is gone.

Two: the identity gap. The AI gets used across three parts of the business. Everyone's pleased. Then somebody asks a simple question: when this thing took an action — moved a record, sent an email, approved a request — who did that? And the honest answer is that nobody can tell. It was all running under one shared login. Its actions sit in the logs mixed in with everyone else's. There's no way to reconstruct who, or what, did what. The fix is not a patch. The fix is rebuilding the whole thing under a proper identity model, which means paying for the build twice.

Three: the human who stopped reading. This is the quietest one. The system was set up so a person reviews every recommendation before anything happens. For the first few weeks, they genuinely read them. By month two, the thing is right so consistently that they start approving in batches. By month four, the review step is a rubber stamp with a person's name on it. Then the system gets one wrong on something that matters, the reviewer approves it without reading, and a client pays for it. Afterward everyone asks how the oversight failed. It didn't fail. It worked exactly as designed. The design just assumed a human would stay interested, and humans don't.

Every one of these is preventable. None of them is prevented by launching well.

The five anchors

Day-Two readiness comes down to five things being true before the system goes live — not after.

THE FIVE ANCHORS OF DAY-TWO READINESS

1. **Know which agent did what.** Every AI process gets a name, an owner, a defined set of permissions, and a limit on how far it can go without asking a person first. This sounds obvious. In most places that have already deployed something, it isn't in place.
2. **Treat your prompts as policy.** The instructions you give an AI aren't settings. They're your business rules, written in English instead of code. They encode how you price, what you'll say to a client, where you draw a line. If anyone with access can edit them, then anyone with access can edit your policy.
3. **Decide where humans stay.** The choice isn't "a person approves everything" or "it runs alone." It's a dial. For each use, you decide where the dial sits and what kicks a decision back to a person — and you design that review so it stays real instead of becoming a rubber stamp.
4. **Measure quality continuously, not once.** Regular software is tested before it ships. AI is different. The system that passed every test on launch day can quietly get worse over the following months. So you check it on every change, and you watch it in the wild.
5. **Build the people who run it.** The people who will operate and oversee an AI system are not the people who built it. They need different skills — supervising it, spotting when it's off, handling the exceptions, managing change. Firms that build that capability early run their AI well. Firms that don't end up with a governance binder that looks great and breaks in practice.

Notice what's *not* on that list: nothing about which model to pick, which vendor to sign, or how much to spend. That's deliberate. Day-Two failures are almost never caused by picking the wrong tool.

A five-question diagnostic

If you can't answer yes with confidence to all five, you have Day-Two exposure. That's not a judgment. It's just a fact about where you are.

FIVE QUESTIONS TO ASK BEFORE THE NEXT ONE GOES LIVE

1. For every AI process running in your business, can you name its owner, its purpose, and what it's allowed to do?
2. Are the instructions guiding it under version control, with someone who reviews changes?
3. Have you explicitly decided, for each use, where a human stays in authority — and designed the work so that person stays engaged?
4. Do you check quality on every change, and watch for drift in normal running?
5. Are the people who will operate and oversee it being trained for that today, before the thing they'll oversee goes live?

Five out of five means you're running an AI program. Fewer than five means you're running a pilot — even if it already shipped, even if it's been live for a year, even if it's working beautifully right now.

That distinction is the whole point. A pilot that's working is not the same thing as a program. It just looks the same from the outside, for a while.

Bottom line

The good news is that Day-Two readiness can be designed in after the fact. Nobody has to rip anything out. But it does have to be someone's job, and it has to be decided rather than assumed.

The work isn't light. It also isn't optional. The firms that build this now will be running real AI at scale a year from now, and earning the trust, the speed, and the audit-readiness that comes with it. The firms that don't will spend that year fixing things.

Plan for Day Two before Day One, and Day One stops being the interesting question.

This note is adapted from The Day-Two Problem: What Most Companies Miss After They Launch AI*, an executive whitepaper co-authored with Richard Mohrmann of TechMohr LLC. The longer paper covers the enterprise and regulated-industry version of the same problem.*

Where your operation sits on Day-Two readiness is one of the things the ISA diagnostic surfaces. Take the 90-second version at **isaadvisory.com →**