THOUGHT LEADERSHIP

Why Transformation Platforms Get Abandoned (And What the Ones That Stick Do Differently)

Most of what goes wrong after a transformation platform goes live has nothing to do with the software. An implementation lead on what the deployments that stick do differently: process first, a named admin, a pilot group, and letting the PMO own training.

Vassily Yasmine September 14, 2026

I run implementations. That means I'm usually the person on the call six months after go-live when a customer says the platform isn't being used, and I've noticed something about those calls: almost none of them are about the software.

Looking back over the last couple of years, something like 80 to 90 percent of what gets raised after a transformation platform goes live is about engagement and process. Who's supposed to update what, when, and what happens if they don't. Actual technical problems are a small minority. But the two get lumped together, and a program that hasn't sorted out its governance ends up blaming the tool.

So here is what I've seen separate the deployments that stick from the ones that quietly drift back to Excel.

START HERE

Decide the process before you configure anything

The first document I ask for isn't a list of features. It's the process guideline: how often initiative owners update progress, who reviews it, what counts as proof that a deliverable is done, and who signs off.

One transformation office I work with runs a monthly cycle. Initiative owners update their tasks through the month and attach evidence. At month end the transformation office reviews. If something's not satisfactory it goes back to the owner; if it is, it goes to the sponsor for validation; then the office formally closes it. Everything in the platform is configured to support that sequence. Notifications, permissions, the review views, all of it.

The same customer has a second program running in the same platform with a completely different rhythm, because it's an integration program where risks move weekly and a monthly cycle would be too slow. The process is different, so the setup is different. If you configure first and work out the process later, you end up with a tool that fights the way the team actually works, and the team wins that fight every time.

OWNERSHIP

Name an admin and a sponsor on day one

Every deployment that has gone well has had a named administrator on the customer side from the first week. Not a committee. One person, ideally in the PMO, who owns user access, the configuration request process, and the relationship with us.

Every deployment that has struggled has had that role unfilled or spread across three people who each assumed someone else was doing it. The admin doesn't need to be technical. The advanced stuff, workflow changes and the like, comes to my team through a request tracker anyway. What they need is to be the person the organization goes to first.

The executive sponsor matters for a different reason. When an initiative owner has skipped two update cycles, the reminder that gets them moving doesn't come from the software.

ROLLOUT

Onboard a pilot group, then everyone

I always recommend onboarding a small group first. Pick one pillar or one workstream, ideally with a couple of people who are reasonably comfortable with tools, and run them through the setup for two or three weeks. Not as a test of the platform, more a check of whether the process holds up with real people and real data.

You will find things. Permissions that are too broad or too narrow. A field nobody understands. A notification that fires at the wrong moment. Fixing those with ten users is a conversation. Fixing them with two hundred users is a rollout problem, and it costs you trust you won't get back easily.

One customer put it well when we were planning this: better to refine the details with a pilot than onboard everyone and have a big challenge in place. That's exactly it.

TRAINING

Let the PMO own training

We'll build the training pack and we'll be in the room, but I push for the customer's PMO to actually deliver end-user training rather than us. Partly because they know the program and can tailor the examples. Mostly because it changes where people go with questions afterwards. If the vendor trained them, they email the vendor. If the PMO trained them, they go to the PMO, which is who should be answering “how do we do X here” questions anyway.

We stay available as the technical advisor. Users can reach me directly if they need to. But the ownership sits with the customer, and that's a large part of why it lasts.

AUTOMATION

Turn the notifications on

This sounds trivial and it's one of the most common gaps I see. The platform can remind an owner that a task is due, tell a reviewer something is waiting, flag an overdue item to the sponsor. All of that only happens if someone switches it on and points it at the right people.

Notifications are what let the process run without the PMO chasing people manually. Without them, the PMO is back to sending emails, which is the job the platform was supposed to remove. There are templated ones that can be live on day one. Start with those and tune later.

PERMISSIONS

Give people only what they need to see

A dashboard that shows an initiative owner the entire program is overwhelming and slightly pointless. They need their tasks, their milestones, their risks. The same dashboard should show the transformation office everything.

Permissions do this. Normal users can create tasks and milestones. Admins can create initiatives and workstreams. Some workstreams are visible to some groups and not others. Getting this right isn't about security so much as about making the platform feel manageable to each person who opens it.

MAINTENANCE

Don’t make the PMO rekey data

If a plan changes, or someone leaves and their forty tasks need reassigning, the PMO should not be editing forty records by hand. That is how a platform becomes a burden. Bulk upload and bulk update through a spreadsheet template should be part of the admin training from the start, so the PMO can maintain the plan themselves without waiting on the vendor.

The one caution is that bulk operations need validation. Make sure whatever you use checks the file before it executes, and only change the rows you mean to change.

DIAGNOSIS

Separate bugs from engagement, and be honest about which is which

When a program is early and there have been data issues, trust is fragile. I've had renewal conversations where the question was really: is the platform stable, or is the team just not using it? Those need different answers.

The way I try to frame it is that we never hit a milestone where everything is done. High engagement drives continuous improvement, and continuous improvement drives engagement. Most of what comes up after the first couple of months is the second kind of thing, and it should be treated as normal, not as evidence that something is broken.

THE RECAP

What this looks like in practice

Eight things that separate the deployments that stick

WRITE THE PROCESS FIRST	Write the governance process down before any configuration.
NAME THE OWNERS	Name the admin and the sponsor.
PILOT BEFORE SCALING	Pilot one group for two to three weeks, then scale.
PMO TRAINS, VENDOR ADVISES	PMO delivers training; vendor advises.
TURN ON NOTIFICATIONS	Turn on notifications and point them at the right people.
SCOPE THE VIEWS	Scope what each role sees.
TEACH BULK UPDATES EARLY	Teach the admin bulk updates early.
TRACK SEPARATELY	Track bugs and engagement issues separately.

None of this is specific to our platform. It's what I'd say to anyone rolling out a strategy execution tool of any kind. The software is a fraction of the outcome. The rest is whether the organization has decided how it wants to run the program and put the people in place to make that happen.

If you're planning a rollout and want to talk through the setup, book a demo and ask for me.

Frequently asked questions

Why do transformation management platforms fail to get adopted?

Usually because the governance process wasn't defined before the tool was configured. In our experience the large majority of issues raised after go-live are about engagement and process, not software defects. If nobody has decided how often owners update, who reviews and who signs off, the platform ends up fighting the way the team works.

Who should administer a strategy execution platform on the customer side?

One named person, ideally in the PMO, from the first week. They own user access, the configuration request process and the day-to-day relationship with the vendor. They don't need to be technical; advanced configuration goes to the vendor's implementation team through a request process.

Should you pilot a transformation platform before rolling it out?

Yes. Onboard one pillar or workstream for two to three weeks before opening it to everyone. The pilot checks whether the process, permissions and notifications hold up with real users and real data. Fixing problems with ten users is a conversation; fixing them with two hundred is a rollout failure.

Should the vendor or the PMO deliver end-user training?

The PMO, with the vendor present as technical advisor. The PMO knows the program and can tailor examples, and more importantly it means users bring questions to the PMO afterwards instead of to the vendor, which is where process questions should be answered.

How do you tell whether a platform problem is a bug or an adoption issue?

Track them separately from the start. Technical defects are usually a small minority of what gets raised after the first month; most items are requests to adjust process, notifications or views as the team's way of working matures. Treating those as normal continuous improvement, rather than as evidence the tool is broken, is what keeps trust intact.