

The handover nobody scheduled between pilot and production

Every pilot has a kickoff and a closing review. Almost none has a day on which another team takes over, and that is the day the work either continues or stops.

Author	Amit Govil, Founder and CEO, XONIK
Published	October 22, 2026
Reading time	8 min read
Online	https://xonik.com/insights/perspectives/the-handover-nobody-scheduled-between-pilot-and-production/

IN BRIEF

Nobody takes over an AI pilot, because the date on which someone should is on no project plan. The team's brief ends on a date, while the system keeps running. What has to pass across is mostly judgment, not code, and the receiver should do the work, with the team still present, before the team leaves.

KEY FINDINGS

1. The handover happens by default, as a side effect of people leaving.

Access is withdrawn when the contract ends and the team's channel goes quiet, while the system carries on producing answers. The day another group becomes answerable is something the team never needs, so it is never scheduled.

2. A repository, a deck and a login are not a handover.

Without the knowledge that a cleaning step existed, why it was needed and how anyone would learn it had stopped working, the receiver cannot tell whether the model, the data or the contracts changed. The cost is the trust of the people who were meant to use the tool.

3. The receiver proves a handover by doing the work while the team watches.

The receiver makes a real change, releases it by the written route and deals with a fault the team has planted. A rehearsal that finds no gaps is also a result, and it is the only evidence the handover has happened.

4. A dated takeover with a named receiver makes the rest possible.

The receiver needs time, access and the authority to change or pause the system without asking the team that built it, and the sponsor supplies whatever is missing. The same date applies when a vendor runs the system.

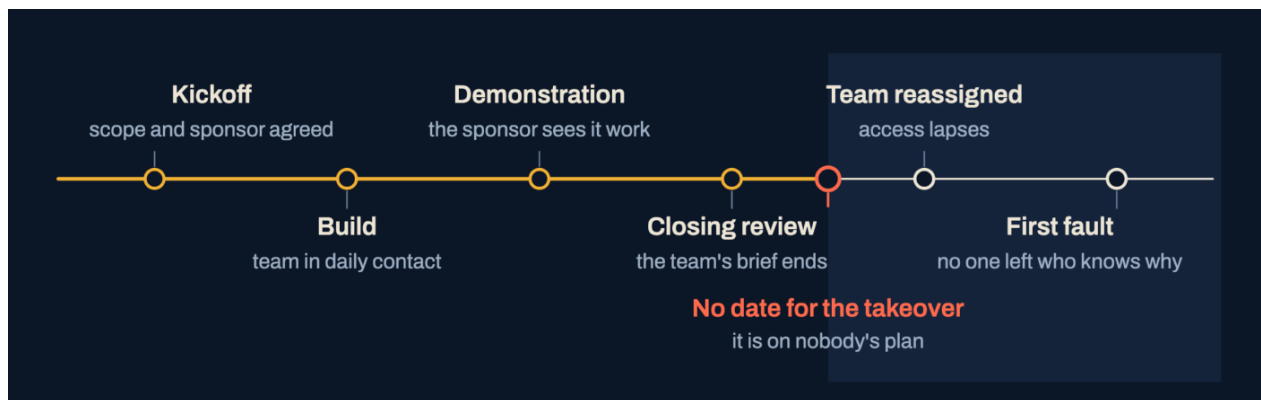
Who takes over an AI pilot, and on what date?

Nobody does, because the date is on no one's plan. A project plan lists what the team needs to finish: the build, the demonstration, the closing review, the final invoice. The day on which another group becomes answerable for the system is not something the team needs, so it is never scheduled. The sponsor assumes the function that will use the system has it in hand. The function assumes the sponsor arranged something. The platform group has heard of the pilot and has not been asked.

The handover therefore happens by default, as a side effect of people leaving. Access is withdrawn when the contract ends. The team's channel goes quiet. The system carries on producing answers, because nothing about a model stops when a budget does. Exhibit 1 sets the project's calendar against what the system needs, and marks the point at which the plan stops naming a date.

EXHIBIT 1

The plan stops naming dates when the team's brief ends



Every date on the plan is one the team needs. The one the system needs is not there.

What does the first week without the project team look like?

Take a legal department that has piloted a model to sort incoming contracts by type and flag unusual clauses. The pilot worked on exports that a paralegal prepared by hand. The closing review went well. On the last day, the department's operations lead receives a repository link, a slide deck, a recording of the demonstration and an administrator login. On any reasonable reading, the work has been handed over.

Some weeks later the contract management system is updated and one of its export fields is renamed. The model keeps running. For one type of contract it now finds fewer unusual clauses, because the field it leaned on arrives empty. A lawyer notices that the flags look thin and asks whether the tool is still any good. The operations lead cannot tell whether the model has degraded, the data has changed or the contracts have. The person who chose how the exports were cleaned is on another engagement and answers messages when time allows. The reply arrives after the lawyers have decided the tool is unreliable and gone back to reading everything.

Nothing in the repository was wrong. What was missing was the knowledge that a cleaning step existed, why it was needed, which field mattered, and how anyone would find out when it stopped working. That knowledge lived in the team's heads, and it left with them. The cost is not the repair, which takes an

afternoon once someone knows where to look. It is the trust of the people who were meant to use the tool, which does not come back on the same terms.

What actually has to be handed over?

A folder holds the easy part. Code, the model or the configuration of a vendor product, credentials and documentation can be copied, checked and signed for, and a handover that stops there looks complete on paper.

The rest is knowledge and standing. Operating knowledge: what the system does badly, which inputs defeat it, what healthy data looks like and what unhealthy data looks like, and the small routine that keeps it running. Decisions and their reasons: the exceptions agreed with the business, the approaches tried and dropped, who approved what. Without the reasons, a receiver will reverse a decision unaware of what it protected, or defend one without being able to say why. Relationships: who uses the output, who supplies the data, who must be told when the output is wrong, who approves a change. And authority: the right to change the system, pause it and switch it off without first asking the team that built it.

Little of this was ever written down, because the team never needed it written down. It lived in conversations held in the same room. A handover is the work of turning conversation into something another person can use, and the only reliable way to find out what is still missing is to watch the receiver try.

Can the receiver do these without calling the team?

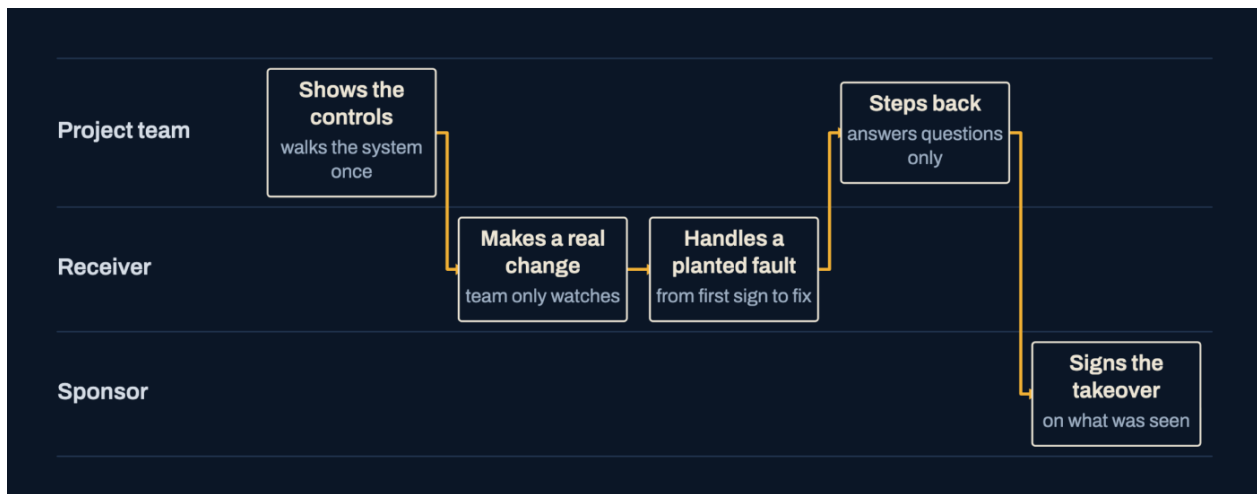
- Say in plain language what the system is for and where it is known to be weak
- Say where each input comes from and what a healthy input looks like
- Find out within a working day whether a wrong answer came from the model, the data or the process
- Make a small change and release it by the written route
- Pause the system and bring it back
- Name who must be told when the output is wrong, and tell them

| The only reliable way to find out what is missing from a handover is to watch the receiver try.

How do you rehearse a handover?

Reverse the roles. For a period before the team's brief ends, the receiver drives and the team watches. The team walks the receiver through the controls once. Then it hands over the keyboard: the receiver makes a real change and releases it by the written route, then deals with a fault the team has planted, such as a renamed field in the input, from the first sign through diagnosis to a fix. The team answers questions and does nothing else. Exhibit 2 shows who does what, including the sponsor's part, which is to sign the takeover on what was seen and not on a document.

EXHIBIT 2

A rehearsal puts the receiver at the controls

The takeover is signed on what the receiver was seen to do, not on what was handed over.

What makes the rehearsal count?

The rehearsal counts only when the brief and the calendar support it. The team's brief has to say that it ends when the rehearsal is passed and not when the build is finished, so that a team that has stopped building has not stopped working. And the overlap has to be on the calendar and in the budget from the start, because an overlap added late is the first thing a slipping schedule removes.

A rehearsal that finds gaps has done its job. The date moves, and the gaps are closed while the people who know the answers are still present. A rehearsal that finds none is also a result, and it is the only kind of evidence that the handover has happened.

What changes the outcome?

Put the takeover on the plan as its first milestone, dated, with the receiver's name beside it, before the build begins. A name, not a role and not a team: a role can be vacant and a team can disagree about who answers. The receiver needs time in the calendar, access to the systems, and the authority to change or pause the system without asking permission of those who built it. If any of those is missing, supplying it is the sponsor's job, which is why the sponsor attends the last session.

The same date applies when a vendor runs the system. The receiver is then the person who will manage the vendor: who knows what the contract covers, what the vendor's support will and will not do, and how to tell whether a change on the vendor's side has altered the behavior. A bought system has a handover too, and the one who receives it is the buyer.

A dated takeover also disciplines the pilot itself. The team starts writing for a reader. The scope shrinks to what a named person can carry. Questions that production would have asked anyway arrive while they are cheap: where the data comes from, who is told when the output is wrong, how the system is stopped.

The public frameworks point the same way without mentioning pilots. The NIST AI Risk Management Framework 1.0 asks that post-deployment monitoring plans are implemented, including mechanisms for

incident response, change management and decommissioning (MANAGE 4.1), and that roles and responsibilities are documented and clear to the individuals and teams concerned (GOVERN 2.1). A rehearsal is the cheapest way to learn whether those mechanisms exist in practice, because it exercises each of them once, with the people who understand them still in the building. NIST describes the framework as voluntary.

None of this is measured. It is what delivery work shows: the systems that carried on after the team left were the ones whose receivers had already done the work, and the ones that stalled had been handed a folder.

Sources

The framework was read on 20 October 2026. The contract scenario and the rehearsal are patterns drawn from delivery work. They are not accounts of a single engagement.

1. [NIST, Artificial Intelligence Risk Management Framework 1.0 \(publication AI 100-1, January 2023\): the subcategories MANAGE 4.1 and GOVERN 2.1, and the framework's description of itself as voluntary](#)

USE THIS

[Assess one pilot with the readiness assessment](#)

[Read the research on stalled pilots](#)

Put the handover on the project plan as its first milestone, with the receiver's name beside it.

This document was made from the article at

<https://xonik.com/insights/perspectives/the-handover-nobody-scheduled-between-pilot-and-production/>. The online article is the current version.