

Why AI pilots stall before production

What fails is seldom the model. The work stops where one team passes a proven model to another that must run it, because nobody owns that moment. This report traces the mechanism, what it costs and how to design the hand-off before the pilot is built.

Author	Amit Govil, Founder and CEO, XONIK
Published	October 22, 2026
Reading time	29 min read
Online	https://xonik.com/insights/reports/why-ai-pilots-stall-before-production/

IN BRIEF

AI pilots stall because a pilot proves a model can do a task under shortcut conditions, while production needs live data, approval from the people who answer for risk, a place in the working process and an owner. The four gaps meet at the hand-off, which nobody owns. Designing it before the pilot is built closes them.

KEY FINDINGS

1. A pilot's shortcuts keep what production needs out of view

A clean extract, volunteer users, a run beside the working process and approvals set aside each remove a delay. The closing report speaks for the extract and says nothing of the conditions it left out.

2. Nobody owns the hand-off, so the stall is quiet

The project team, the business function, technology and risk each hold a piece of the system's life, and the move between them is in none of their plans. No one decides to stop the pilot. It stops being reviewed.

3. Live data, late approvals and an unchanged process each turn a working pilot into rework

Production needs a feed, not a file. Approvers who first see the design at go-live ask questions that cost a rebuild, and a process left as it was pays the full manual cost and then pays to review the model as well.

4. An owner is someone who can change, fund, monitor and stop the system, not a name on a slide

A pilot never needed an off-switch, because the team simply stopped it. At go-live nobody inherits that power, so wrong outputs continue, or the system cannot be retired.

5. The signs of a coming stall are visible while the pilot runs, and none concerns the model

A leader can check them in short conversations without asking the project team to prepare anything. A single sign proves little, and signs that appear together describe a pilot whose hand-off has not been

designed.

6. Decisions taken at the start, each with a person and something that exists afterward, change the outcome

The pilot can still take shortcuts if each one is written down with what it hides and when it will be removed. If no owner or no function will take the system, stopping before building costs less than a pilot that stalls.

What does this report ask, and why does it matter now?

This report asks why a pilot that has shown a model can do a task can still stop before the task is done in daily use, and what a leader can change before the pilot is built. It is written for the executive who sponsors a pilot, inherits one, or will be asked to approve one: a head of AI, a chief information or operating officer, the head of the function whose people will use the output, and the risk, security and compliance leads who answer for what goes live.

The short answer is that the model is seldom what fails. A pilot proves a model can do a task under shortcut conditions: a clean extract of data, volunteers as users, a run beside the working process, and approvals set aside because the work is only a test. Production needs four things the pilot never had. It needs data that arrives continuously. It needs approval from the people who answer for risk, security and compliance. It needs a place inside the working process where the output is used, with the process changed to match. And it needs an owner. Exhibit 1 shows them as layers beneath a working model.

EXHIBIT 1

Four things sit beneath a working model

○ A working model	what the pilot showed	The pilot never needed these
○ An owner	decides, notices, can switch it off	
○ Approvals	risk, security and compliance have seen the design	
○ A place in the process	the output is used, and the work around it has changed	
○ Live data	arrives continuously, with gaps and late corrections	

A pilot shows the top layer. The four below it are what production adds, and each is supplied by a different person.

Where do the four needs meet?

They meet at the hand-off from the project team to the people who must run the result. The project team owns the pilot, the business function owns the process, technology owns the platform and risk owns the approvals. The hand-off belongs to none of them, so nobody is assigned to carry it out, and the stall that follows is quiet. It costs the pilot budget. It also costs the credibility of the next proposal, which is harder to

approve because the last one never arrived.

Three things make the question pressing. A pilot can be started with little more than a model, an extract and a sponsor, so a business can have more than one waiting at the hand-off at once, each with its own project team and none with a route to production. Public audit bodies now ask for what production needs: the UK National Audit Office, reviewing the use of AI in government in March 2024, listed clear accountabilities, outcome measures, workforce impact, legacy systems and data, and the right skills among the lessons to get right at the outset (paragraph 3.8). And regulation has begun to name the same things. The EU AI Act sets duties for deployers of high-risk systems in Article 26, and a regulation of 8 July 2026 moved the dates from which they apply. That gives time to prepare and no reason to wait.

[Our briefing on AI regulation and enterprise deployments](#)

Method and limits

This report is analysis from delivery work. It rests on two things. The first is the delivery experience of the firm since its founding in 2006, described as a pattern and a mechanism, because every engagement is under NDA and no case can be shown. The second is a set of public documents, each opened and read for the provisions cited, on 20 October 2026: the NIST AI Risk Management Framework 1.0 (NIST AI 100-1, January 2023) and the NIST AI RMF Playbook; Regulation (EU) 2024/1689 and the amending Regulation (EU) 2026/1744; the UK National Audit Office report Use of artificial intelligence in government (March 2024); and the UK Government's Artificial Intelligence Playbook (February 2025).

No chart is drawn. The data we hold counts the enterprises that use AI, not how pilots move into use, and a figure that does not answer the question would only decorate it. Each finding takes one mechanism: why it happens, what you can see, and what it costs. The public documents are used for what they ask of an organization that runs AI and for what they say about how a test differs from live use. None of them says that pilots stall.

- It does not say how often pilots stall, or how often each of the four gaps is the cause. It states no rate, share or count, because there is no sample behind one.
- No client work appears. We describe a pattern, and a reader should test it against their own pilots instead of taking it on our word.
- The National Audit Office report and the UK Playbook concern public bodies. They are cited for what they say about ownership, data, assurance and process change, not as evidence about private companies.
- The NIST framework is voluntary. The EU provisions apply to high-risk systems as the Act defines them; for any other system they are a reference for good design and not a duty. Nothing here is legal advice.
- The EU Regulation was read as published in the Official Journal on 12 July 2024, and the amending regulation of 8 July 2026 was read for what it changes. It moves dates of application and replaces Article 72(3). It does not amend Articles 14 and 26. The dates may change again.

What is a pilot built to prove?

A pilot is built to answer one question quickly and cheaply: can a model do this task? Every shortcut the project team takes removes a source of delay. A clean extract removes the work of connecting to the live system. Volunteers remove resistance and the need to train a whole function. Running beside the process

removes the need to change it. Setting approvals aside, because the work is only a test, removes the wait for the people who answer for risk.

Each choice is sound for the question being asked. Together they decide what the pilot can show. Exhibit 2 sets the pilot's conditions against those of production, and its last column says what each shortcut keeps out of view.

EXHIBIT 2

Pilot conditions and production conditions

Condition	In the pilot	In production	What the pilot cannot show
Data	A clean extract, cut once	A source that delivers continuously, with gaps and late corrections	How the model behaves on records nobody cleaned
Users	Volunteers who forgive errors	Everyone whose job includes the task, doubters among them	Whether the output is trusted without its champions
Place in the work	Beside the process, as a second view	Inside the process, where the output changes what happens next	Whether the process can absorb the output
Approvals	Set aside because it is a test	Required from risk, security and compliance	Whether the design would be allowed to run
Ownership	The project team, until the brief ends	A named person who answers for the system after launch	Who decides, who notices, who can stop it
A wrong answer	A person notices and shrugs	A wrong output enters a record or a decision	What a mistake costs
Support	The builders on call	A run team working from a run book	Whether anyone but the builders can fix it

Each shortcut suits the pilot's question. The last column names what it hides from the people deciding on go-live.

How do the public documents describe the difference?

The public documents describe the same gap between a test and live use. The NIST AI Risk Management Framework notes that measuring AI risk in a laboratory or a controlled environment may give useful insight before deployment, and that the measurements may differ from the risks that emerge in operational, real-world settings (section 1.2.1). It adds that accuracy figures should be paired with realistic test sets that represent the conditions of expected use (section 3.1). The NIST Playbook, in its guidance on MEASURE 2.3, says that testing confined to test data sets does not directly evaluate risks and impacts in real-world environments, and it suggests measuring systems before deployment in conditions like the expected ones and documenting the differences between the measurement setting and the deployment environment.

A published public-sector example shows the shape. The UK Government's AI Playbook describes a procurement recommendation tool that a team first built as a pilot on historical data, served by email and in conversation because no online platform was ready to carry it. Once the platform was ready, the Playbook says, new challenges appeared, among them keeping recommendations flowing without a break and measuring how people interacted with them in the service. Exhibit 3 shows the general path from a successful pilot to that point.

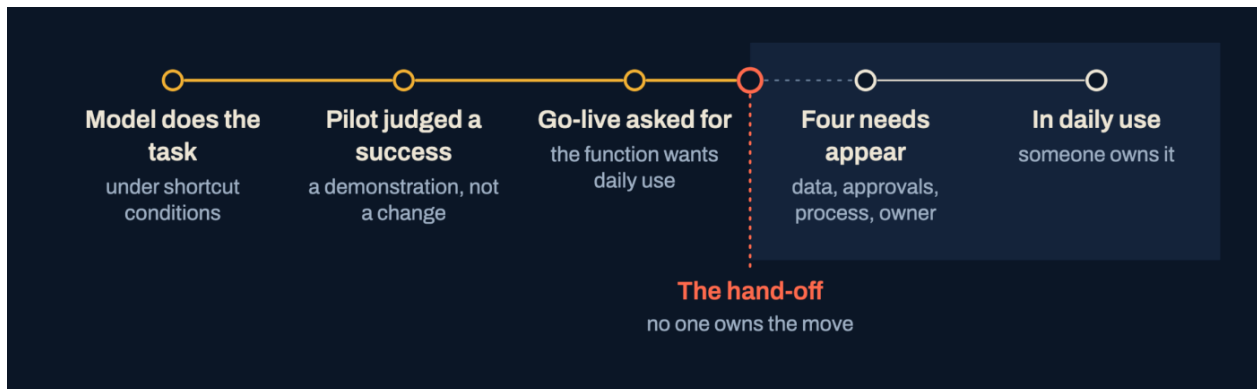
[NIST AI 100-1: Artificial Intelligence Risk Management Framework \(AI RMF 1.0\), NIST, January 2023](#)

[NIST AI RMF Playbook, NIST AI Resource Center](#)

[Artificial Intelligence Playbook for the UK Government, GOV.UK, 10 February 2025](#)

EXHIBIT 3

From a successful pilot to the hand-off



The line is unbroken until go-live is asked for. Past that point, four needs arrive together and no one has been assigned to meet them.

What does the narrow evidence cost?

What you can see is a closing report that gives results from the extract and says nothing of the conditions it left out. The sponsor hears that it works. The budget conversation that follows is about scaling a result that has not yet met the conditions it will scale into.

The cost arrives late and all at once. The work the shortcuts postponed becomes visible together, at the point when the project team is being released and the sponsor expects a date. Because nobody listed the shortcuts, nobody can say how much work they postponed, and the first estimate is a guess made under pressure.

Why does nobody own the hand-off?

Four groups each hold a piece of an AI system's life. The project team owns the pilot until its brief ends. The business function owns the process the output would enter. Technology owns the platforms and the people who run them. Risk, security and compliance own the approvals. The move from the first group to the other three is in none of their plans, because it is not a deliverable of the project, a budget line of the function, a service of technology or a review step of risk. It is the gap between four reasonable jobs. Exhibit 4 shows who holds the work at each step, and where the line to the next holder is missing.

The NIST framework describes the general condition. In its Map function it observes that people in charge of one part of the AI lifecycle often lack full visibility of, or control over, other parts, so that good intentions in one dimension can be undermined by decisions and conditions in later activities (section 5.2). Its Appendix A is more specific. It lists the tasks of deployment as including checking compatibility with legacy systems, ensuring regulatory compliance, managing organizational change and evaluating user experience, and it names operators, end users and system integrators as the people who carry them out. A pilot's brief is

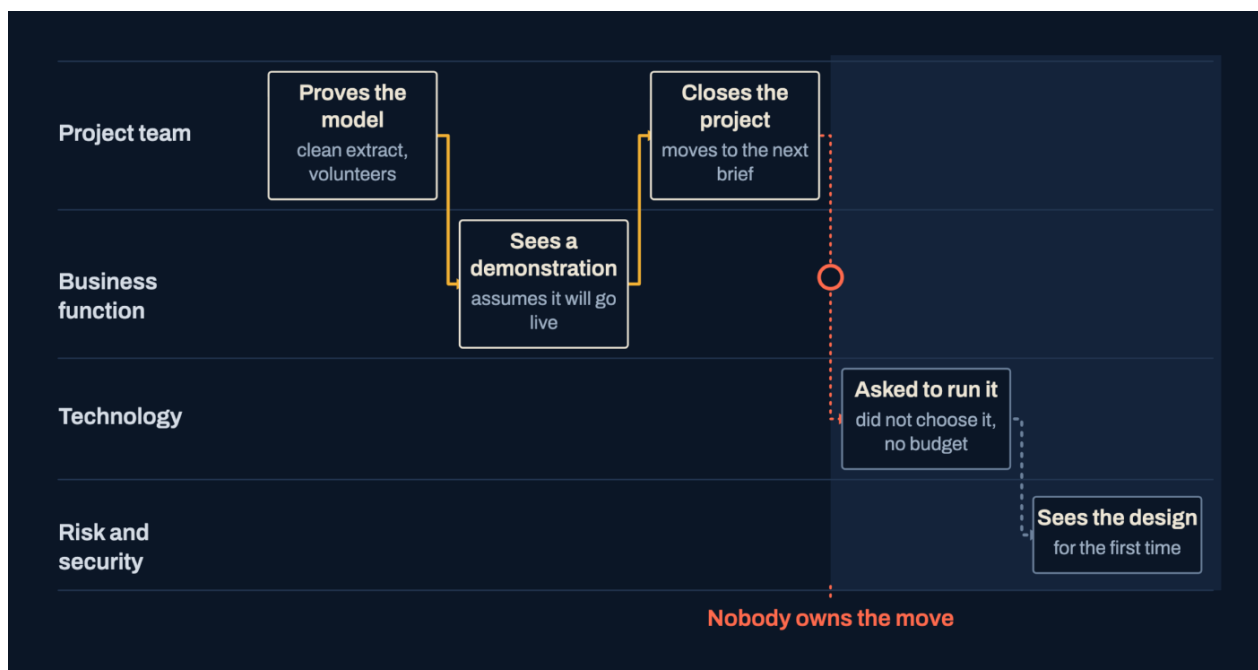
written around the model, so those tasks fall to whoever arrives when go-live is requested.

The National Audit Office found the same structure at the scale of a government. Responsibility for AI adoption sat across more than one body, with potential for overlap, and the draft strategy did not say which held overall ownership and accountability for delivery (paragraphs 9 and 3.8). The UK Playbook's advice on building a team is the practical form of the same point: scaling and deploying a solution brings in software and infrastructure teams, and engaging them early gives the project clear requirements and sets a path for success.

[Use of artificial intelligence in government, National Audit Office \(HC 612\), March 2024](#)

EXHIBIT 4

Who holds the work, and where nobody does



The first three steps follow in order, and then the work passes to groups that were never given it. The hand-off is the one move with no lane.

The hand-off is the gap between four reasonable jobs, and it is in none of their plans.

How does the stall happen, and what does it cost?

What you can see is a sequence in which each party behaves sensibly. The project team has finished its brief and is wanted on the next one. The business sees a demonstration and assumes technology will make it real. Technology sees a model it did not choose and has no budget to run. Risk has not been asked. No one decides to stop the pilot. It stops being reviewed, the sponsor's attention moves, the budget year ends.

The cost has three parts. The pilot spend is lost. The knowledge the project team carried, such as why the data was cleaned in a certain way and which inputs the model handles badly, leaves with the team. And the next proposal meets a more skeptical room, because the last one did not arrive. The third is the hardest to see on any budget line.

Why does a model that worked on a clean extract struggle on live data?

An extract is a snapshot, selected and cleaned by people who know the source. That work hides what production data is like: records joined by hand, fields whose meaning changed over the years, corrections that arrive weeks late, entries made differently by different teams, and access that someone had to grant. The cleaning steps are also part of the system's performance. If the model only works after a person fixes the input, the fixing is logic that has to be built, tested and owned.

Production needs a feed, not a file. That means a named owner of the source who has agreed to supply it continuously, checks at the point of entry, a record of which data the model saw when it gave an answer, and a way to notice when the data changes shape. The NIST Playbook, in its guidance on MEASURE 2.4, describes why: systems encounter new issues in production as the environment changes, an effect it calls drift, so that they no longer meet the assumptions of the original design. Its suggested actions include monitoring how production metrics differ from those collected in pre-deployment testing, and setting alerts for when the distribution of new input data differs from the test outcomes. The framework itself lists a need for more frequent maintenance because of data, model or concept drift among the ways AI systems differ from traditional software (Appendix B).

The National Audit Office makes the same point about public bodies. It reports that significant quality problems in operational data hinder transformation, and that limited access to good-quality data was identified as a barrier to implementing AI (paragraphs 3.12 and 3.13). It also warns that building AI tools on top of existing systems, without addressing the underlying issues, limits how much of the potential can be used (paragraph 3.10). For a deployer of a high-risk system, Article 26(4) of the EU Act adds a duty: to the extent the deployer controls the input data, it must be relevant and sufficiently representative for the intended purpose.

What you can see is accuracy reported on the extract that nobody can reproduce on the live source, a refresh that is a manual task done by the one person who understands it, and a first live feed that exposes empty fields and values the model never met. The cost is time at the worst moment, or a worse outcome: a go-live without discovery, in which wrong outputs enter records quietly and are found by the people downstream.

Why do approvals arrive after the build?

A pilot avoids approval by being declared a test. The people who answer for risk, security, privacy, legal exposure and compliance see the design when go-live is requested. They then ask questions that the design cannot answer without rework: where the data goes, what is logged, who can see the output, how a person can override it, what happens when it fails. These are legitimate questions. Asked at design, they cost a decision. Asked after the build, they cost a rebuild, and each change restarts a review.

The public guidance is consistent that this should happen early. The UK Playbook tells teams to connect with the right assurance teams early in the project life cycle and to document review and escalation processes (Principle 10), and it states that a service must comply with the government's Secure by Design principles before it can be deployed. The National Audit Office found that public bodies named legal risk, the risk of inaccurate outputs and security risk among the barriers to implementing AI, and wanted support from the center on them (paragraphs 3.16 and 3.17).

Article 14 of the EU AI Act shows what such questions look like when written as design requirements. A high-risk system must be designed so that people can oversee it while it is in use. The people assigned to oversight must be enabled to understand the system's capacities and limits, to remain aware of the tendency to over-rely on its output, to interpret it, to decide not to use it or to override or reverse it, and to interrupt it through a stop button or similar procedure that brings it to a halt in a safe state (Article 14(4)). Article 26 then makes the deployer responsible for assigning that oversight to people with competence, training and authority and the support they need (26(2)), for monitoring the system and suspending its use where it may present a risk (26(5)), for keeping the logs under its control (26(6)), and for informing workers and their representatives before a high-risk system is used at the workplace (26(7)). A stop procedure and a log cannot be added by the last week of a build. They are features of the design. The NIST Playbook, on MAP 3.5, goes further: it says to include the relevant people in prototyping and testing, and to evaluate oversight procedures before deployment where the setting is high-stakes.

What you can see is a review meeting that is the first time risk has seen the system, followed by a list of questions, each of which changes the design, each change restarting the cycle. Reviewers who were not involved default to caution, which is rational for them. The cost is months on the calendar after the builders have gone, and a standing wariness toward AI proposals in the functions that must approve them.

[Our briefing on AI regulation and enterprise deployments](#)

What happens when the process around the model stays the same?

A pilot shows its output in a separate view and leaves the work as it was. In production, the output has to be used at a defined point by a defined person, and the steps around it have to change to match. Where the model takes over a step, the neighboring steps change: who checks, who handles the exceptions, which record is the authoritative one, what the performance measure now counts, and what the people who did the old step do instead.

When none of this is done, two routes persist. The manual route continues because nobody has set a date on which it stops. The output turns into a rival record, kept alongside the real one. People keep private copies, consult the model when they remember, and trust it when it agrees with them. The National Audit Office states the principle: adoption plans ambitious enough to deliver the benefits envisaged require changes in technology together with significant business process and workforce changes (paragraph 3.7), and it lists assessing workforce impacts among the lessons to settle at the outset (paragraph 3.8). It also notes, from a case study, the importance of capacity within operational teams to trial AI use cases and support adoption.

The oversight guidance describes the human side. The UK Playbook says human input must be meaningful, and advises teams to design, document and assess the stages at which human review takes place and what further information the reviewer will weigh. Article 14(4) of the EU Act names the opposite risk, a tendency to rely automatically on output. In an unchanged process both failures are likely: some people ignore the output because nothing makes them use it, and others defer to it because nothing tells them when to challenge it. The NIST Playbook, on MAP 3.5, recommends differentiating the people who oversee a system from those who use it.

What you can see is two numbers in the same meeting, a tool used only by those who sponsored it, exceptions with no route, and a performance measure that has not moved. The cost is that the gain the pilot pointed to is not realized, because the process still pays the full manual cost and now also pays to review the

model. Use declines, and the verdict that AI does not work here follows, though the model was never given a place to work.

Who owns the system after the project team leaves, and who can switch it off?

In a pilot the owner is the project lead, and the lead's authority ends with the brief. At go-live nobody inherits it. Ownership has distinct parts that are easy to blur: the authority to decide when the system changes, the budget to run it, the duty to read its monitoring, the route by which an incident is reported and answered, and the power to stop it. A name on a slide supplies none of these. The off-switch is the part easiest to forget, because a pilot never needed one: when it misbehaved, the team simply stopped it.

The frameworks treat this as basic. The NIST framework asks that roles, responsibilities and lines of communication for mapping, measuring and managing AI risk are documented and clear (GOVERN 2.1), that executive leadership takes responsibility for decisions about the risks of developing and deploying AI (GOVERN 2.3), that monitoring and periodic review are planned with roles defined (GOVERN 1.5), and that there are processes for decommissioning systems safely (GOVERN 1.7). Under MANAGE it asks for post-deployment monitoring plans that include appeal and override, decommissioning, incident response, recovery and change management (MANAGE 4.1), and for mechanisms, with responsibilities assigned and understood, to supersede, disengage or deactivate systems whose performance is inconsistent with their intended use (MANAGE 2.4). The Playbook's guidance on MANAGE 2.4 adds that decision thresholds for bypass or deactivation belong to continual monitoring, and that safe removal may need backup systems so that work can continue.

The UK Playbook says to nominate a senior responsible owner accountable for the use of AI in a specific project, to set out how the model will be maintained and managed over time with a plan for knowledge transfer, to state who has authority to change the model's code, and to have fallback processes so that critical services continue if a change is reverted or a system is terminated. The EU Act puts the same ideas in law for high-risk systems, in the stop procedure of Article 14(4)(e) and the deployer's duty in Article 26(5) to suspend use where a system may present a risk.

What you can see is a live system with no name beside it in the inventory, a monitoring view that nobody opens, a vendor model that can change without notice to anyone, and silence when someone asks who can switch it off and what happens to the work if they do. The cost is that wrong outputs continue, or that the system cannot be removed and is kept running because nobody can authorize otherwise. When it is finally retired, it happens without a plan. The NIST Playbook warns that irregular or indiscriminate termination or deletion of a system can raise risk, because it may be subject to regulatory requirements or to later legal and security investigations (GOVERN 1.7).

What does a stall look like before it happens?

The signs are visible while the pilot is running, and none concerns the model. Exhibit 5 lists the signs we look for, what each usually means, and where a leader can check it without asking the project team to prepare anything. Read the list as questions rather than a score: a single sign proves little, and signs that appear together describe a pilot whose hand-off has not been designed. Each one names a person who can confirm it in a short conversation.

EXHIBIT 5

Signs of a coming stall and what each means

Sign	What it usually means	Where to look
No name sits beside the system after launch	The owner is assumed, not appointed	The project plan and the AI inventory
Success is defined as a demonstration	The pilot will end when the demonstration does	The charter and the sponsor's own words
Data was checked only on the extract	The live feed has not been tried	The data engineer's notes
Risk and security have not seen the design	Approvals will start after the build	The meeting list and the review log
The process around the model is not drawn	The output has no place to land	The process map, or its absence
Nobody can say what happens when the model is down or wrong	There is no fallback and no off-switch	The run book, or its absence
The manual route has no end date	Two sources of truth will persist	The function head's plan
The project team is already booked on the next work	The hand-off day is in nobody's calendar	The resource plan

Every sign can be checked from documents that already exist, which is why a leader can check them without a project review.

What is the question nobody asks?

One further sign is a question that is not asked: what happens when the model cannot be reached, or gives a wrong answer? A team that cannot answer it has a pilot, not a design for production. The answer is not mainly technical. It names a person, a threshold, a fallback and a way to tell the people who used the output that it was wrong.

The pattern changes when the hand-off is designed before the pilot is built, not after it succeeds.

What changes the outcome?

The pattern changes when the hand-off is designed before the pilot is built, not after it succeeds. That means five decisions taken at the start, each with a person who takes it and something that exists once it is taken.

First, name the owner and give them a say in the design. The sponsor takes this decision. What exists is a name, a statement of what the person is responsible for and can decide, and a budget line for running the system after launch. Second, draw the route to production. The owner takes it with the project lead, technology and risk. What exists is a one-page route: the sources the data comes from, the systems the output enters, the approvals required and who gives them, and what is monitored by whom. Third, change the process, not only the software, and set the date on which the manual route stops. The head of the function takes this decision. What exists is a process map with the model's place on it, and a date.

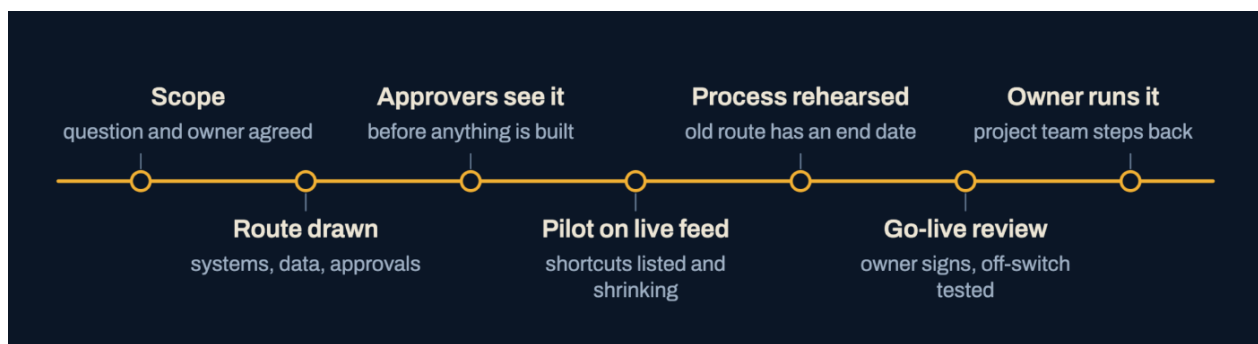
Fourth, agree what happens when the model is down or wrong. The owner takes this with technology. What exists is a fallback, the threshold at which someone acts, the person who can stop the system and the way the people who used its output are told. Fifth, agree the conditions under which the system will be retired. The owner takes it with risk. What exists is a short statement of what would end the system's life, and what is kept when it does.

The pilot itself can still take shortcuts. The change is that each is written down, with what it hides and when it will be removed. That list is the most useful document a pilot produces, because it is the work the hand-off will have to do. If, when the route is drawn, no owner will take the system or no function will change its process, the right outcome is to stop before building, which costs less than a pilot that stalls.

None of this slows a pilot in a way that matters. It moves forward the questions production would ask anyway, to a stage when the answers cost a design decision and not a rebuild. It is also what the public documents ask for: the NIST Playbook, on MAP 3.5, asks that the relevant people take part in prototyping and testing under scenarios similar to deployment conditions, and the National Audit Office's lessons begin with the business need and clear accountabilities. Exhibit 6 shows the order of events when the hand-off is designed first.

EXHIBIT 6

A hand-off designed before the pilot is built



Every question production would ask is asked before the build, and the project team leaves only after the owner has run the system.

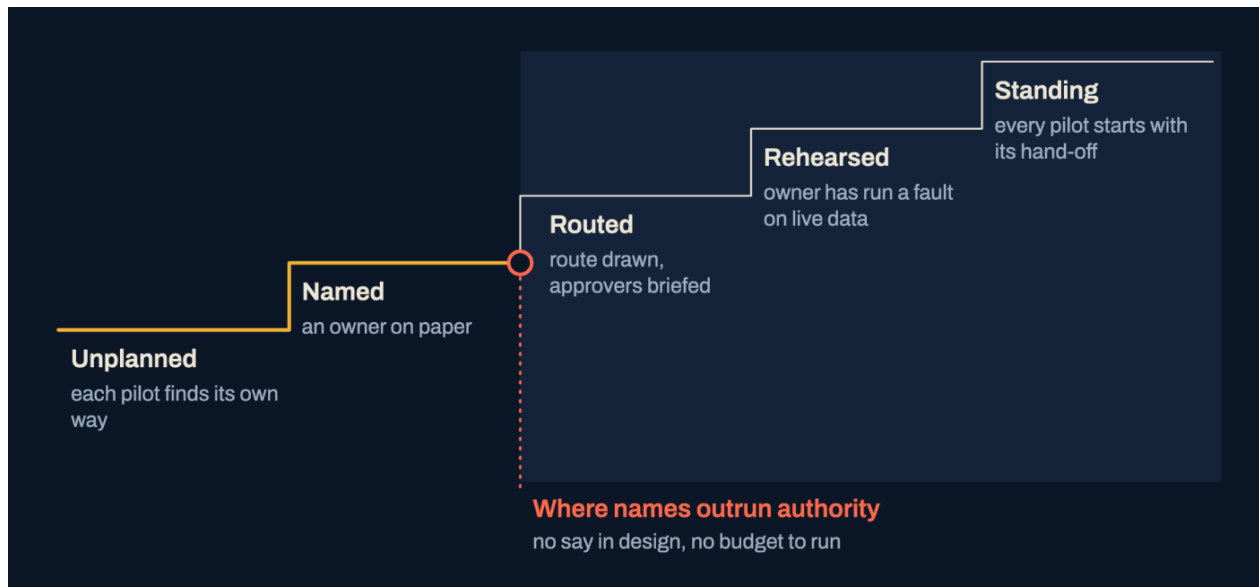
How ready is an organization's hand-off?

Organizations differ in how reliably they cross the hand-off, and the difference can be read from what they can produce on request. Exhibit 7 sets out five levels. At the first, unplanned, each pilot finds its own way and the outcome depends on who happens to care. At the second, named, an owner appears on paper. A name is cheap and reassuring, and authority is not, so the test is whether the owner had a say in the design, holds a budget to run the system and can stop it.

At the third level, routed, the route to production is drawn and the approvers have seen it, so the questions that arrive after go-live have already been asked. At the fourth, rehearsed, the owner has run a change and handled a fault on live data before the project team leaves. At the fifth, standing, the hand-off is designed as a matter of course when a pilot is chartered, and systems that are retired are recorded with the reasons. The NIST framework points the same way: GOVERN 1.6 asks for mechanisms to inventory AI systems, and GOVERN 1.7 for processes to retire them safely.

The way to place a pilot, or an organization, is to ask for the document each level implies: the owner's written responsibilities, the one-page route, the approvers' written conditions, the record of a drill, the register of retired systems. A level that cannot be evidenced has not been reached.

EXHIBIT 7

Maturity of the hand-off

The step from named to routed is where the work happens: it turns a name into authority, and a hope into a route.

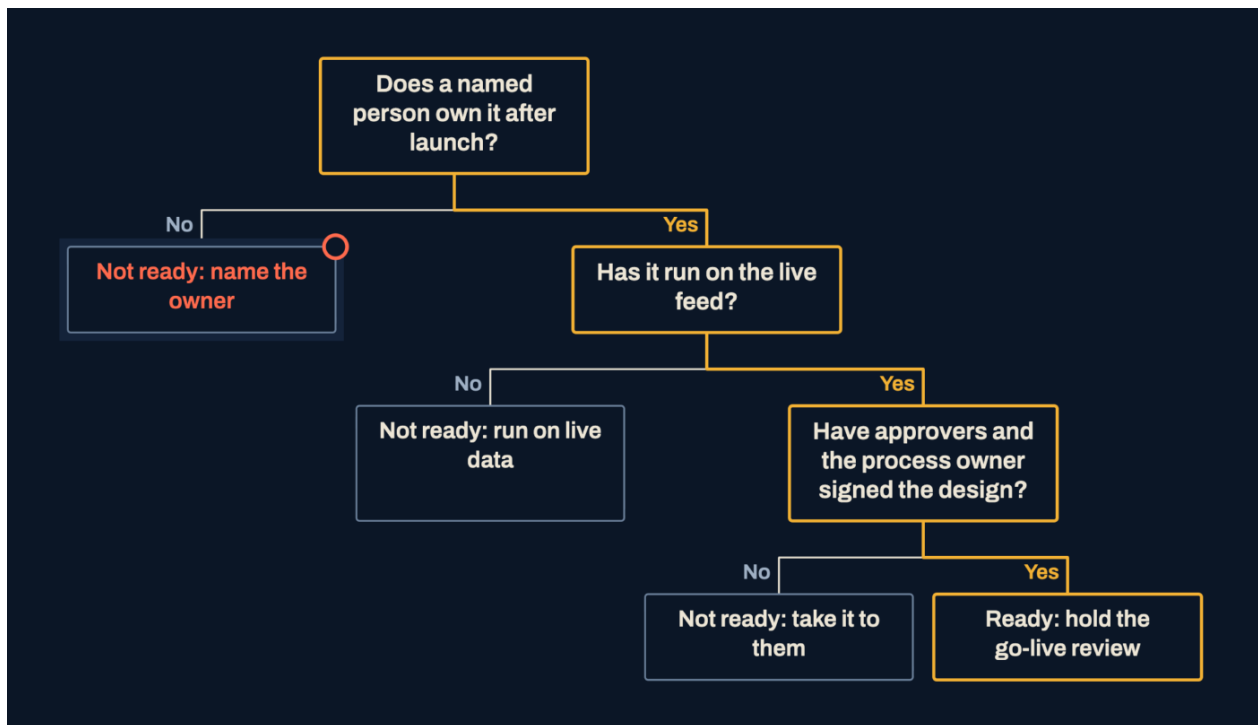
How do you decide whether a pilot is ready to go live?

A leader can make the decision with four questions asked in a fixed order. The order matters. The owner comes first, because without one nobody can answer the later questions or act on the answers. The live feed comes second, because running on real data is what exposes the design changes that approvers need to see. The sign-off of the design comes third, from the risk and security approvers and from the owner of the process that will change, since they must agree to the same design. If all three answers are yes, the pilot is ready for a go-live review, not yet for go-live.

Exhibit 8 shows the sequence. A no at any question has its own remedy, and the remedy is not to push on. A pilot that fails the first question is not at the hand-off yet; it is before it, and the first job is to name the owner.

EXHIBIT 8

Is this pilot ready to go live?



Three yes answers earn a go-live review. Any no names the next piece of work.

What is the step-by-step response?

1. List the pilots in flight

The sponsor and the head of technology list every pilot, its stage and who will own it after launch. NIST GOVERN 1.6 asks for an inventory of AI systems, and the UK Playbook asks for a live one. At the end there is an inventory with a column for the owner and a column that says whether a route to production exists.

2. Decide which pilots to carry forward

The sponsor decides, pilot by pilot, to continue, to hold or to stop. A pilot with no one willing to own it stops or holds. At the end there is a short list with the reasons written down, so a dropped pilot can be revisited.

3. Name the owner and give them authority

The sponsor and the head of the function name one person for each pilot that continues, and agree what that person can decide, spend and stop. At the end there is a written statement of responsibility and a budget line for running the system.

4. Draw the route to production

The owner, with the project lead, technology and risk, draws one page: the data sources and who supplies each, the systems the output enters, the approvals needed, the monitoring and its reader. At the end there is a route that every party has read and none has disputed.

5. Take the design to the approvers

Security, privacy, legal and compliance review the design with the owner, before more is built. They say what they need to see: logs, access, human review, a stop procedure. At the end there are written conditions for approval and a list of the changes they imply.

6. Run it on the live feed

The owner and a data engineer run the model on the live source, beside the existing process if needed, and compare its behavior with the pilot's. At the end there is a record of where the two differ, and the monitoring that will catch drift.

7. Change the process

The head of the function and the process owner redraw the work around the model: who acts on the output, who handles exceptions, which record is authoritative, what is measured. Everyone who will use it is trained, not only volunteers. At the end there is a new process map and a date on which the manual route stops.

8. Write the run book and test the off-switch

The owner and technology write the run book: monitoring thresholds, who acts at each, how an incident is reported and answered, how the system is stopped and restarted, what the fallback is, and what would retire it. They then run a drill. At the end there is a run book and a record that the drill was done.

9. Hold the go-live review

The sponsor chairs a review with the owner, the approvers and the function head. The owner presents the route, the live results, the conditions met and the fallback. At the end there is a signed decision to go, to hold or to stop.

10. Watch the first full cycle, then step back

The project team stays through one full cycle of real use while the owner runs the system and makes the first change. Then the team is released. At the end there is a short review note, an owner who has run the system, and a pilot that has become a service.

How can a leader check a live pilot today?

- I can name the person who answers for this system after the project team leaves, and that person knows it.

- That person had a say in the design, holds the budget to run it and has the authority to stop it.
- Success is defined as a change in how work is done, not as a demonstration.
- The model has run on the live source, not only on an extract, and the differences between the two are written down.
- The source of the data has an owner who has agreed to supply it continuously.
- Risk, security, privacy and compliance have seen the design, and their conditions are written down.
- People who did not volunteer have used the output.
- The process map shows where the output is used and who acts on it.
- The manual route has an end date.
- There is a written answer to what happens when the model cannot be reached and when it gives a wrong answer.
- Someone reads the monitoring on a set rhythm, and a threshold exists at which they act.
- Someone with authority can stop the system, and a drill has shown that it can be done.
- The conditions for retiring the system are agreed.
- The day the project team hands over is in a plan, and the team stays until the owner has made a change.

What do readers ask about stalled AI pilots?

Who is this report written for?

It is written for the executive who sponsors a pilot, inherits it or will be asked to approve it: a head of AI, a chief information or operating officer, the head of the function that will use the output, and the risk, security and compliance leads.

Does the report say how often AI pilots stall?

No. It rests on delivery experience described as a pattern, plus public documents read for the provisions cited, and it states no rate, share or count because no sample stands behind it. No client work appears, so test the pattern against your own pilots.

How can I tell that a pilot of mine is heading for a stall?

The signs are visible while the pilot runs, and none concerns the model. The report lists them with what each usually means and who can confirm it in a short conversation. A sign alone proves little, but signs that appear together describe a pilot whose hand-off has not been designed.

What should be decided before a pilot is built?

Name an owner with a say in the design, draw the route to production, change the process and set a date when the manual route ends, agree what happens when the model fails, and agree when it retires. The pilot can still take shortcuts if each is written down.

Sources and notes

Every document was opened and read for the provisions cited on 20 October 2026. NIST AI 100-1 is dated January 2023, and NIST states that a revision is under way; the Playbook will be updated after it. The EU Regulation was read in the Official Journal text of 12 July 2024 and in the amending regulation published on

24 July 2026. The UK Playbook was published on 10 February 2025 and the National Audit Office report on 15 March 2024.

1. [NIST AI 100-1: Artificial Intelligence Risk Management Framework \(AI RMF 1.0\)](#), NIST, January 2023. Read: sections 1.2.1, 1.2.4, 3.1 and 5.2, Appendix A and Appendix B, and the subcategories GOVERN 1.5, 1.6, 1.7, 2.1 and 2.3, MAP 3.5, MEASURE 2.3 and 2.4, and MANAGE 2.4 and 4.1.
2. [NIST AI RMF Playbook](#), NIST AI Resource Center. Read: the entries for GOVERN 1.7, MAP 3.5, MEASURE 2.3, MEASURE 2.4 and MANAGE 2.4.
3. [Regulation \(EU\) 2024/1689 \(Artificial Intelligence Act\)](#), Official Journal, 12 July 2024. Read: Articles 14 and 26.
4. [Regulation \(EU\) 2026/1744 \(Digital Omnibus on AI\)](#), Official Journal, 24 July 2026. Read: the changes to dates of application and to Article 72(3).
5. [Use of artificial intelligence in government](#), National Audit Office (HC 612), 15 March 2024. Read: paragraphs 9, 1.10, 3.7, 3.8, 3.10, 3.12, 3.13, 3.16 and 3.17.
6. [Artificial Intelligence Playbook for the UK Government](#), GOV.UK, 10 February 2025. Read: Principle 10 and the sections on governance, human oversight, security, building the team and its case studies.

USE THIS

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

[Use the production readiness playbook](#)

If a pilot of yours is at the hand-off now, name its owner this week. The rest of the work gets easier once someone has the authority to do it.

This document was made from the article at <https://xonik.com/insights/reports/why-ai-pilots-stall-before-production/>. The online article is the current version.