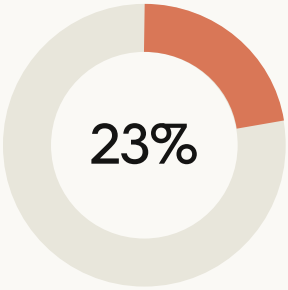
Deploying AI from pilot to production

A PRACTICAL BLUEPRINT FOR CIOS AND TECHNICAL LEADERS

ANTHROP\C | accenture

Table of Contents

Rising to the demands of the AI era



Percentage of C-suite leaders who have achieved sustained, enterprise-wide AI impact

Accenture, Pulse of Change, July 2026

Ambition has outrun readiness

64%

Have moved beyond pilots into production across multiple functions, or initiated enterprise-wide efforts for advanced AI

7%

Have reached the data-readiness required to scale it

Accenture, AI-Ready Data for Advanced AI survey, May 2026

Enterprise AI has moved past experimentation. The organizations deploying it at production scale are seeing returns that would have seemed implausible two years ago and that keep accelerating as model capabilities evolve.

A global insurer compressed its underwriting review timeline by more than 5x while improving data accuracy by 75 to 90%. A pharmaceutical company cut clinical study report production from ten weeks to under ten minutes. A telecom provider deployed more than 13,000 custom AI solutions across its workforce in under a year.

These are production deployments running on real data, at scale. And as frontier models and technologies become increasingly more capable, the door to even more advanced and impactful use cases is wide open.

Prior enterprise technology cycles took years to mature, but AI is compressing that timeline into months or even weeks. The organizations that successfully build and deploy these capabilities now accumulate compounding advantages that become increasingly difficult for late movers to close.

Many enterprise AI programs haven't reached this point. According to Accenture's July 2026 Pulse of Change report, only 23% of C-suite leaders report having achieved sustained, enterprise-wide AI impact with AI initiatives. Readiness to scale shows a similar shortfall: 64% of respondents to Accenture's May 2026 AI-Ready Data for Advanced AI survey say they have moved beyond pilots into production across multiple functions or initiated enterprise-wide efforts for advanced AI, yet only 7% have reached the data-readiness required to scale it. Moving successfully from pilot to an organization-wide rollout is where many programs stall.

Closing the gap between pilot and production

Pilots often succeed because they aren't bound by the same rigorous criteria of production-scale deployments. They operate with protected budgets, purpose-built teams, curated datasets, and highly technical or AI-native engineers who are not representative of the broader enterprise workforce.

They also have limited exposure to the stakeholders and departments that will eventually own whatever gets deployed.

That insulation makes pilots successful, but it also makes them an unreliable signal for how enterprise AI will perform in production at scale. The fix is to incorporate the governance, operational ownership, delivery stability, and economics as a part of the pilot design itself.

That means:

- Defining production success criteria upfront
- Testing against representative enterprise conditions
- Minimizing hidden manual intervention
- Validating delivery scalability beyond AI-native specialist teams
- Embedding governance, observability, and operational controls early
- Measuring long-term operational economics, not just short-term model performance

As a result, organizations run a pilot that functions as a production readiness exercise rather than just a demonstration of technology capabilities.

This guide outlines seven considerations that can help AI programs successfully reach production. Each consideration highlights the operational, organizational, and technical decisions that must be addressed before moving forward.

Each consideration also includes two types of questions:

- **Work out** questions require cross-functional input and have multiple owners
- **Assign** questions are ownership decisions that a CIO or business leader must make

The organizations that get AI to meaningful production treat their pilots with the same diligence they apply to real-world deployments, from success criteria to stakeholder involvement and infrastructure decisions. The seven considerations that follow are drawn from what we've observed across the organizations that made it to production: Anthropic through enterprise deployments built on Claude, Accenture through the implementations it has guided across industries and geographies.

Consideration 01

PRE-PILOT

Strategic objectives, ROI, and ownership

Build your evaluation framework before the pressure builds

AI pilots tend to leave success criteria loosely defined, for understandable reasons. The goal of a pilot is to demonstrate that a technology can do something useful, not to specify the exact conditions under which it should be deployed at scale.

The problem is that many pilots operate under artificially optimized conditions: curated data, AI-native engineering teams, narrow scopes, protected budgets, and a high level of manual intervention. Those conditions rarely represent the realities of enterprise production environments.

As organizations move from experimentation to deployment, ambiguity around success criteria becomes a major risk. The questions shift from "Can we model work?" to:

- Can it operate reliably across fragmented enterprise systems and data?
- Can standard delivery teams support it at scale?
- What governance and oversight are required?
- Are the economics sustainable in steady-state operations?

Set the performance bar upfront

When organizations haven't established and agreed on success criteria before production, different stakeholders arrive with different definitions. For example, in an AI-assisted contract review deployment, success might mean something different for each stakeholder:

- The Engineering team wants to track if the system flags non-standard clauses accurately enough that attorney review time drops below what manual review required.
- The Finance team wants to track if the cost per reviewed contract, accounting for attorney time on flagged items, is lower than the fully-manual process.
- The Legal/Risk team wants to track if every output carries a complete audit trail that satisfies their documentation requirements.

Success criteria that don't converge pre-pilot are unlikely to converge post-launch. Instead, they are likely to generate competing narratives about whether the program is working.

Define your use case

Establishing success criteria before a pilot begins does two things: it sets the human performance baseline the AI will be measured against. It also forces a conversation about what's actually being measured. A well-defined use case is, arguably, the most critical part of your evaluation framework.

When drafting your use case definition, be as detailed as possible and include a defined user, task, output, and a measurable quality threshold.

> "The AI can analyze documents" is a capability, not a use case. → "The AI reviews lease abstractions for the acquisitions team and flags non-standard clauses for counsel" is a use case.

The more precisely a use case is scoped before an AI pilot begins, the easier it is to set the right success criteria and identify the data and integration requirements the production deployment will depend on.

How StubHub defined AI success before shipping

StubHub, an Anthropic customer, conducted rigorous A/B testing with multiple AI models before committing to a production deployment, measuring against specific resolution rate and satisfaction benchmarks rather than launching on general performance impressions. The result was a 30% reduction in support costs and response times that dropped from over 20 minutes to near-instant.

> "The decision to choose Claude was entirely data-driven. We tested multiple model providers side by side, and Claude consistently delivered the best results for case resolution rates and customer satisfaction scores."
>
> — Timothy Addison, Engineering Org Chief of Staff, StubHub

Tie success criteria to ROI before the pilot begins

Success criteria and ROI are closely related, but organizations often treat them as separate questions addressed at separate times. The programs that scale well treat them as the same question, answered pre-pilot.

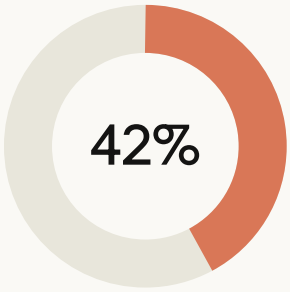
Start with high-frequency, high-value workflows where ROI compounds. A process that runs once a quarter and takes two hours to complete isn't the right candidate for an AI deployment with significant integration overhead. A process that runs hundreds of times daily is a much better candidate, as the economics scale differently.

Additionally, build a lightweight total cost of ownership model before the pilot begins: API costs, integration overhead, maintenance burden, and the

opportunity cost of not deploying. Then revisit it quarterly as AI economics evolve. Model costs have dropped significantly over the past two years and architectures built around last year's or even last month's pricing assumptions may be over-engineered for today's.

Finally, when defining your success criteria, cover both leading indicators and lagging indicators of success, as well as the lag between the two. For example, a leading indicator for value derived from a developer using Claude Code may be a decrease in PR cycle times. But that doesn't directly correspond to business value. The lagging indicator, whether that's increase in revenue, improved net revenue retention, reduced churn, or something else, should also be tracked and attributed to the new features or products being brought to market.

Establish clear ownership



Percentage of organizations that rely on shared IT and finance accountability, with no single owner responsible for AI costs and outcomes

Accenture, Tokenomics research, September 2026

Someone has to be accountable for holding the program to its success criteria, and that owner needs to be named before the pilot begins. Accenture's September 2026 Tokenomics research found that 42% of organizations rely on shared IT and finance accountability with no single owner responsible for AI costs and outcomes. Without clear ownership, it becomes difficult to measure success, make tradeoffs and maintain accountability as programs scale. That may be an individual or a small team with a named lead, but the accountability has to be unambiguous.

The owner will have decision rights over product choices, the authority to escalate blockers, and executive backing when organizational friction surfaces.

- **Decision rights** enable the owner to make calls without convening a meeting and incurring the delays that follow
- **Escalation authority** ensures that blockers requiring senior attention get resolved in days rather than months
- **Executive backing** signals to the rest of the organization that the program has institutional weight, which determines whether other functions engage as partners or as observers

Decision rights, escalation authority, and executive backing aren't interchangeable, and the absence of any one of them is enough to stall a program.

Document the governance process

Accountability at the top matters, but so does the governance structure that surrounds it. The stakeholders who will eventually approve, own, or operate the deployed system need to understand how decision-making and escalation works.

That means documenting the go/no-go process before the pilot begins:

who has decision rights, what criteria trigger escalation, how the program communicates wins and setbacks, and what the rollout sequence looks like. For organizations that don't define this upfront, these early-stage inputs can turn into late-stage blockers.

Broad cross-functional input can improve decision quality, but input and progress require different structures. Build a governance model that gives stakeholders a clear process to trust and a clear owner to hold accountable.

Each consideration in this guide ends with two sets of questions. 'Work out' questions require cross-functional input and don't have a single owner. 'Assign' questions are ownership decisions that a CIO or business leader must make before the program advances.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
What metrics define success for this use case, and how will they be measured?	Who owns evaluation design and scoring methodology?
What is the human performance baseline, and how was it established?	Who reviews the TCO model as AI economics change?
What accuracy or quality threshold justifies moving from pilot to production?	Who is the executive sponsor, and what is their active commitment?
What are the go/no-go criteria and who evaluates them?	Who owns the escalation path when progress stalls?

Consideration 02

PRE-PILOT

Data readiness and integration

Know what production data looks like before the pilot begins

Enterprise AI pilots tend to use curated, clean data from a single system that someone prepared for that specific exercise. Production AI deployments run on the real data estate, which is often distributed across systems built at different times, formatted inconsistently, subject to governance requirements that weren't designed with AI in mind, or owned by teams whose priorities don't align with a deployment timeline.

Between the pilot and production, data causes more programs to stall than almost any other factor. Accenture's AI-Ready Data for Advanced AI survey found that only 7% of organizations have reached the level of data-readiness required to scale advanced AI, including generative, agentic, and physical AI. Those that do, "data reinventors," realize EBIT margin uplifts of up to 1.6x over industry peers. The gap between having data and having AI-ready data is technical and strategic, and is often discovered only after committing to a production timeline.

7% of organizations have reached the level of data-readiness required to scale advanced AI

Those that do, "data reinventors," realize up to **1.6×** EBIT margin uplift over industry peers

Accenture, AI-Ready Data for Advanced AI survey, May 2026

Three problems stall most programs: access and quality issues in the data, integration that takes more engineering than the pilot scoped for, and infrastructure built for earlier model limits that no longer apply. Each is cheaper to fix before the timeline is locked, which is why teams that catch all three early are more likely to reach production on schedule.

How Novo Nordisk treated data governance as a precondition to deployment

Novo Nordisk, an Anthropic customer, operates under strict data governance requirements for clinical trial documentation and patient data. Before building NovoScribe, their AI-powered documentation platform, the team established exactly what data could flow to and from the model and what controls had to be in place first. As a result, clinical study report production dropped from more than 10 weeks to under 10 minutes.

> "In a highly regulated industry, we can't just throw our data into a large language model and hope for the best. Our conversations with Anthropic guided us on how to securely use Claude for planning, strategic tasks, and code generation."
>
> — Waheed Jowiya, Digitalization Strategy Director, Novo Nordisk

Audit the data before setting timelines

The first place programs stall is at the data audit stage. In 2025, Gartner predicted that organizations will abandon 60% of AI projects unsupported by AI-ready data through 2026.[1]

The problem is that many organizations audit their data after defining the pilot scope and setting the launch date. By then, every issue the audit surfaces delays the program.

When teams do run the audit early, they typically find two types of data issues, both of which are addressable before the pilot begins:

- **Access:** the data exists but can't be reached programmatically, or reaching it requires permissions that take months to navigate through IT and Legal
- **Quality:** the data is accessible but inconsistent, incomplete, or formatted in ways that produce unreliable outputs

Access or data quality problems that would have taken weeks to clean during the pilot can take months to fix under production pressure.

Scope infrastructure and data integration work fully

Once the data is clean and accessible, the second layer surfaces: getting it to flow in and out of production systems is harder engineering work than accounted for in most pilots.

Integration work is consistently where programs discover scope that wasn't in the budget. Pilots tend to run on manually prepared data passed into the model as context, while production deployments require that data to flow bidirectionally, reliably, and at volume across systems that weren't designed with AI in mind. The engineering effort to build that plumbing and the coordination overhead required to navigate IT change management and vendor API timelines almost always exceeds initial estimates.

Data integration is how AI outputs re-enter existing systems and how those systems feed context back to the model. That loop is what determines

[1] Q&A with Roxane Edjlali, Lack of AI-Ready Data Puts AI Projects at Risk, February 26, 2025. GARTNER is a trademark of Gartner, Inc. and/or its affiliates.

whether a deployment sustains itself in production or stays dependent on manual workarounds.

For example, a contract drafting tool drawing from a live clause library is more useful and less risky than one operating on whatever context was manually pasted in. These integrations require understanding existing system architecture, coordinating across IT change management, and in many cases starting vendor API conversations several months earlier than the team running the pilot typically thinks to initiate them.

Reevaluate retrieval architecture as model capabilities expand

Early enterprise AI infrastructure was often built around small context windows that couldn't process long documents in a single pass. Retrieval pipelines and chunky architectures were the workaround: breaking documents into smaller pieces, summarizing before sending, staging data in layers before it reached the model.

That constraint has largely been addressed. Modern models can process entire contracts, codebases, or research reports in a single pass. Before investing in retrieval infrastructure, it's worth auditing whether the architecture is solving a current problem or one that's already been addressed. Retrieval layers still add genuine value when data freshness or access control are the primary drivers. But many existing pipelines are solving for a context window constraint that no longer applies.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
Where does the data the AI needs actually live and what is its current state?	Who owns each data source and pipeline in production?
What access and governance requirements apply to each data source?	Who owns IT change management for new integrations?
What data quality standard is required for reliable AI outputs?	Who is accountable if a data quality issue surfaces in production?

Consideration 03

PRE-PILOT

Infrastructure and architecture

Commit to your AI infrastructure before the pilot begins

When enterprises can't settle on whether to build their own AI infrastructure or use a managed service, they often do both. A pilot runs on a managed API while a platform team builds something in-house for *"when we need more control."* Six months later, the organization is maintaining two systems. Every new use case reopens the same debate and engineers end up solving the same integration problems in parallel.

Most infrastructure problems aren't caused by choosing the wrong path but by never fully committing to one. Build or buy? Single model or multi-model? Cloud or on-premises? Deferring these decisions creates downstream engineering debt: systems that need maintaining, integration patterns that never standardize, and a platform foundation that keeps reopening to negotiation.

To avoid these challenges, make deliberate infrastructure decisions pre-pilot by addressing three questions:

1. What's the simplest architecture that meets the actual requirements — not the assumed ones?
2. What level of model customization is genuinely necessary versus assumed?
3. What data residency and security requirements will Legal and Compliance actually enforce?

Match the model to the use case

Different models perform differently across use cases. A model optimized for reasoning may not be the right choice for high-volume document processing, and one built for code generation may underperform in customer-facing interactions.

Use cases that scale well start with the model best suited to the task and expand when there is a specific, validated need. As organizations scale AI across functions, they should evaluate where different model capabilities are required and align their architecture accordingly.

Before launching the pilot, find out what each use case requires and determine the minimum architecture that delivers it.

How TELUS built a platform that let the best model win

TELUS, an Anthropic customer, built Fuel iX as a multi-model platform because the scale and diversity of their use cases required it from the start. At 13,000 custom AI solutions across a workforce of tens of thousands, a single-model architecture wouldn't have delivered it. Given the choice across multiple models, Claude became the dominant selection, handling the most complex and creative tasks. The result: 500,000+ hours saved and tens of thousands of custom AI solutions built by employees across the organization.

"When TELUS gave our team access to multiple AI models through Fuel iX, Claude became the overwhelming choice. The platform processes 100 billion tokens monthly, and Claude is the preferred model for our most complex and creative tasks."

— Jaime Tatis, Chief AI Officer, TELUS

Align platform choices to use case requirements

Where the AI runs—cloud, on-premises, or hybrid—shapes what engineers can build on, how fast new model capabilities become available, and what the maintenance and operational overhead looks like at production scale. Organizations with existing cloud provider relationships often move faster through procurement and identity provisioning. Those running sensitive workloads on-premises trade deployment speed for tighter control over their data.

Many organizations use multiple platforms: cloud for some use cases, on-premises for others. That's a legitimate approach when it's deliberate, but teams must ensure they coordinate integration patterns, security reviews, and capacity constraints. Like model selection, platform decisions should be made against documented requirements before the pilot begins.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
Build versus buy: what are the real criteria, including engineering and maintenance cost?	Who owns the integration layer in production?
What are the actual data residency and sovereignty requirements?	Who evaluates new model capabilities as they emerge?
What does each use case require, and which models will be most appropriate?	Who is accountable when model versioning or deprecation affects deployed systems?
Cloud, hybrid, or on-premises: what does each platform enable or constrain?	Who is accountable for the platform decision and its governance implications?

Consideration 04

PRE-PILOT

Security, compliance, and trust

Resolve security and compliance issues before the pilot begins

Security, compliance, and trust each govern a different dimension of the deployment: what data can reach the model, what you're legally allowed to build, and whether the business can rely on what gets built. All three need to be established before the pilot begins.

In healthcare, financial services, insurance, and other regulated industries, AI deployment is a regulatory question as much as a technological one. The programs that treat compliance as a late-stage gate consistently discover during the production build that regulatory requirements reshape the architecture they designed during the pilot. The programs that treat compliance as a design constraint avoid this entirely because they surface those requirements pre-pilot, when changes are decisions rather than delays.

Involve legal, compliance, and security stakeholders before a pilot launches, not at the point when they're being asked to approve something that's nearly complete. This way, you can incorporate their requirements for data handling, audit trails, access controls, and model behavior from the start.

Classify your data before setting the pilot architecture

Data classification means categorizing the information your AI system will handle by sensitivity level (public, internal, customer PII, financial, and clinical, for example) so that the right security controls and compliance requirements can be applied to each category. This exercise will inform both security controls and compliance frameworks, and it needs to happen before the pilot architecture is set.

The answers to questions about output visibility, data transmission permissions, logging access, and retention policies differ depending on whether the system handles public information, customer PII, financial records, or clinical data. Establish these answers pre-pilot, so the architecture that handles each data type differently is designed from the start rather than retrofitted.

Establish security requirements

Before the pilot kicks off, the security team needs to assess your AI solution provider's security posture: how the provider handles data in transit, what encryption standards apply, where data is stored, whether the contractual protections match the organization's requirements.

The same applies to access provisioning. Who can interact with the system? What authentication is required? What happens when an employee leaves or changes roles? AI systems inherit the access control requirements of the data they touch, so those requirements need to be mapped before the pilot is built.

How Palo Alto Networks made security a requirement, not an afterthought

Palo Alto Networks, an Anthropic customer, achieved a 20-30% increase in feature development velocity with Claude. When evaluating AI providers for security-critical coding workflows, they treated each vendor's approach to safety as a primary criterion alongside technical performance.

"Anthropic prioritized safety and security a lot more than other LLMs. They discuss security and safety implications in every meeting. As the largest cybersecurity company, that's a big deal for us."

— Gunjan Patel, Director of Engineering, Palo Alto Networks

Map sector-specific compliance requirements

Regulated industries each bring their own compliance requirements and it's important that deployment teams take them into account when mapping out the security and compliance considerations for their pilot programs. Examples of these industry-specific considerations include:

- Healthcare programs touching patient data need HIPAA-compliant infrastructure and signed business associate agreements before a vendor relationship goes live
- Financial services programs generating customer-facing AI outputs must meet regulations around advice, disclosures, and fair lending, which vary by jurisdiction and product type
- Legal and HR programs handling privileged information or employment decisions that must maintain auditable human oversight, which some jurisdictions require

Oversight design needs to be part of the pilot architecture, not added after the system is ready to go to production.

Build behavioral constraints into the model

Most enterprise QA frameworks were designed for deterministic systems where the same input produces the same output. AI systems don't work that way. AI outputs are non-deterministic, and AI systems can produce different outputs from the same input, based on context, phrasing, and model state.

The discipline required to govern non-deterministic outputs is architectural: building behavioral constraints into how the model operates rather than applying validation layers after the fact. Organizations that try to satisfy compliance requirements through testing alone consistently discover the gap at the worst possible time, under production load, with real data, and a live audit.

Approaches that build behavioral constraints into how models operate give compliance and legal teams something they can audit, document, and verify. The relevant question shifts from whether outputs are generally acceptable to whether model behavior can be explained in a regulatory or legal context.

When data classification and security controls are working as designed and system outputs are consistently reliable, you earn trust. That consistency must be built into the architecture from the start rather than retrofitted after the pilot is running or ready to move to production.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
What data classifications apply to the information this system will handle?	Who owns the security assessment before the pilot architecture is set?
What is the vendor's security posture, and does it meet the organization's requirements?	Who owns the vendor contracts requiring data processing agreements?
What regulatory frameworks govern this use case, and what do they specifically require?	Who is responsible for the audit trail for AI-generated outputs?
What access provisioning and authentication requirements apply?	Who has authority to define and enforce access controls?

Consideration 05

PILOT TO PRODUCTION

Organizational readiness

Prepare the organization before production begins

The conditions that make pilots successful often don't exist in production. Pilot teams are handpicked, often selected for enthusiasm and capability, and work on a defined scope with clear timelines. Pilot budgets are often protected, with insulation from normal organizational dynamics. This makes pilots unrepresentative of the conditions under which production actually runs.

Where a pilot has a single team with a clear objective, production involves multiple functions, including IT, legal, finance, HR, and compliance, each with their own priorities and definition of what "working" means. As we already discussed in Considerations 1 and 2, *Strategic objectives* and *Data readiness,* whether a pilot successfully moves to production can depend on when and how those functions engage with the program, which is frequently determined by decisions made at the top before the pilot begins. A CIO or business leader can shape that engagement deliberately, or discover it reactively at go-live.

The programs that make this transition successfully engage leadership and cross-functional stakeholders at the start of the program and treat their requirements as design constraints. Mapping stakeholders early is the part organizations tend to find manageable. Getting the broader workforce to adopt AI in their day-to-day work—and change how they work because of it—is where most programs struggle.

Scale AI adoption across the organization

Deployment doesn't guarantee adoption. Most organizations find a gap between effective AI users and everyone else, and closing it requires pressure from both directions.

To address this gap, leadership has to make adoption a visible priority. That means clearing time for teams to experiment, recognizing early adopters publicly, and treating AI competency as something the organization is actively building rather than assuming it will emerge on its own. Programs that identify early adopters (champions) and deliberately engineer those

demonstration moments, rather than relying on them to happen organically, close the adoption gap measurably faster.

A portfolio manager showing a compliance specialist how she summarized a 200-page filing in three minutes converts more skeptics than any structured rollout.

The organizations that move adoption from early users to the broader workforce start with willing teams, make results visible early, and find people who can show genuine use and showcase results.

Evolve responsibilities and skills as AI-augmented work spreads across the organization

As adoption spreads through the organization, how people and teams work changes. In deployments that scale successfully, work shifts from executing tasks to overseeing a system: watching whether AI is handling its work correctly, stepping in when it isn't, and identifying where the system is quietly degrading before it becomes a problem.

In that shift, judgment-based skills matter: the ability to identify incorrect outputs, decide which cases require escalation, and detect system drift before it becomes a production issue. These capabilities require deliberate investment in training and role design. Organizations will see the most value when they invest in role clarity early: what each team owns now, what they owned before, and how performance gets measured in the new model.

How NBIM built AI literacy across the entire investment organization

Norges Bank Investment Management (NBIM), an Anthropic customer, manages one of the world's largest sovereign funds. The bank's challenge wasn't finding a model that could handle institutional investment analysis but rather ensuring all employees, from portfolio managers to compliance specialists, could use AI effectively. They developed role-specific training paths, launched an AI Ambassador Network of 50 specialists, and measured adoption continuously. Within two months of deployment, more than 600 employees were active users. Internal surveys now show employees saving more than 20% of their time weekly on AI-assisted tasks.

"We didn't want off-the-shelf solutions – we wanted a strategic partnership with a provider who would understand our specific needs as an institutional investor."

— Stian Kirkeberg, Head of ML and AI, NBIM

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
What is the rollout sequence, and why?	Who has veto power over production deployment?
How do you communicate wins and problems honestly with affected teams?	Who is the executive sponsor and what is their active role?
What reskilling is required and what does it cover?	Who are the internal champions and how are they being supported?
How are role changes being communicated?	Who funds the reskilling investment?

Consideration 06

Governance and risk management

Prioritize human oversight for production systems

Production systems that apply the same level of human review to every AI output, regardless of the stakes, either stall under the review burden or start cutting corners in ways that introduce risk. Production governance requires a tiered approach: fast handling for routine, low-consequence outputs and more scrutiny for the ones that carry real weight.

The deployments that scale effectively match their oversight to the risk level of each output. That means two things in practice: auditing oversight regularly so checkpoints earn their cost and building the monitoring and incident response capability to catch what oversight misses.

Governance frameworks that don't get audited can become "structural drag" where review checkpoints accumulate because they're easier to add than to remove, and because the risk of removing a checkpoint is visible while the cost of keeping it is not. Accenture's September 2026 Tokenomics report finds that organizations with formal chargeback accountability for AI spending link 32 cents of every dollar of AI token spend to a quantified business outcome, six times more than those with no allocation. As AI adoption scales, governance structure becomes the critical lever for tracking value creation and making informed investment decisions.

Below, we share our four-tier framework for developing an oversight model, with example use cases across different industries. These are only suggestions, and how organizations structure oversight will depend on internal policies, regulations, and more. The proposed tiers are flexible, as well: if a process has had a low error rate for a long period of time, say six months or a year, it may be ready to move up to tier 2.

Tier	Oversight model	Example use cases	Review cadence
1. Automated	No human review; output goes directly to workflow	• Summarizing meeting transcripts • Reformatting data between systems • Generating code documentation	Quarterly audit of output quality
2. Sampled	Random subset reviewed on a regular cadence	• Extracting structured data from invoices • Classifying support tickets • Populating CRM fields from call transcripts	A portion of outputs reviewed weekly; start with a higher rate of reviews and adjust as error patterns stabilize
3. Reviewed	Human approves every output before it reaches its audience	• Drafting client communications • Generating financial analysis for external distribution • Producing contract language from templates • Creating marketing content	Every output reviewed before release; audit the review process monthly for catch rate and cost
4. Advisory	AI provides analysis, a human makes the decision and produces the output	• Credit and lending recommendations • Candidate screening in hiring workflows • Clinical findings for physician review • Regulatory filing preparation	Every decision documented with an audit trail; compliance review quarterly

Audit manual review checkpoints

Human review is costly and slows deployments down. It has its place early in production, when organizations add human review at every stage because trust in the system is low and risk tolerance is limited. The problem is that those checkpoints rarely get revalidated as the system matures and demonstrates reliability.

Auditing manual checkpoints periodically in production is basic governance hygiene. Ask these questions for each review step:

- What does it actually catch?
- How often does a reviewer change the output?
- What's the cost per review, and what's the justification for incurring it?

If the catch rate is low and the cost is high, the checkpoint can be automated, sampled, or removed.

Define incident response processes and monitoring

When an AI system produces a harmful output, the response quality depends entirely on whether the incident response process was defined before the incident. Ask these questions:

- Who gets notified?
- What's the remediation path?

- Who has authority to pause the system?
- Who communicates to affected parties?

Organizations that handle incidents well have documented answers to these questions before go-live.

Production AI systems also degrade in ways that periodic reviews miss. Output quality shifts when models are updated or prompts drift from their original intent. Data feeding the system can become stale or fall out of sync with what the model was originally evaluated against. Meanwhile, the real-world process the AI was designed to support keeps changing with new regulations, workflows, or edge cases.

These issues surface gradually, as a slow decline in catch rate or a quiet increase in exceptions that nobody connects to a root cause until the problem is significant. Continuous monitoring catches the problems while they're still manageable. To do so successfully, teams must establish what to measure, set automated alerts for degradation thresholds, and define clear escalation protocols.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
What is the decision taxonomy for categorizing outputs by risk and consequence?	Who gets notified, what is the remediation path, and who has authority to pause the system in an incident?
For each manual review step: what does it catch, how often does a reviewer change the output, and what is the cost?	Who owns continuous monitoring and degradation alerting?
What does continuous monitoring cover, and what thresholds trigger escalation?	Who has authority to remove or automate review checkpoints as the system matures?

Consideration 07

IN PRODUCTION

Scale and evolution

Move from AI assistants to end-to-end automation

Most enterprise AI programs start with copilots: tools that help people work faster. Copilots can offer real productivity gains and for many organizations were the first proof that AI can deliver value in production.

But the throughput ceiling on a copilot is still the person using it. A tool that helps an analyst write reports faster is still bounded by how many reports that analyst can review and approve. The economics shift when AI completes the work itself: an agent that processes invoices end-to-end, escalating only the exceptions, or a system that handles tier-one support tickets from open to resolution, looping in a human only when it's uncertain. At that point, throughput scales with volume, not headcount.

AI can now operate interfaces the same way a person would. Agents can fill forms, navigate systems, and execute multi-step workflows. Most organizations start with assistance because it's lower risk. The next step is to identify the use cases that would scale with end-to-end automation and implement it.

Redefine the human role as AI takes on more of the process

When AI automates a workflow, the human role shifts to setting the boundaries the system operates within and intervening when it reaches its limits. The claims processor who used to review every invoice now audits the system that reviews them.

That shift requires deliberate role design. Before scaling automation, the organization needs to specify what authority the human has to override the system, what the escalation path looks like when the system encounters something outside its boundaries, and how performance is measured when AI is handling most of the volume.

Start simple, scale deliberately

While the case for end-to-end automation is clear, getting there can easily become more complex than it needs to be. The key is to avoid building complex agentic AI systems when a simpler approach might work. A well-designed prompt is fast to test and has predictable failure modes, while an agentic system with multiple steps, tool use, and decision-making is more capable but harder to debug, more expensive to maintain, and fails in ways that are harder to anticipate. Start with the simplest solution that could work, validate it against real performance data, and add complexity only when the simpler approach has demonstrably hit its limits. Anthropic's own guidance on building effective agents makes the same argument.

Organizations that skip this sequence often spend months building sophisticated infrastructure for problems that a cleaner, more constrained deployment could solve.

Understand how the system fails before scaling it

An error mode is a specific, recurring category of failure — a type of mistake the system makes. Understanding which error modes exist and how they're distributed at current volume tells you whether the system is ready for more.

A system that's 95% accurate at limited volume sounds production-ready. But the remaining 5% matters. If the typical failure is a formatting issue that a reviewer catches in seconds, full automation may be safe. If the typical failure is an incorrect financial calculation or a missed compliance flag, every additional unit of volume adds risk. Beyond determining the error rate, you must ask, "what does each type of error cost at full production volume?"

Getting specific about error modes before scaling is what separates organizations that expand with confidence from those that discover problems at volume.

Build an ongoing deployment capability

AI capabilities are shifting fast enough that a system optimized for the current model will likely need revisiting in a matter of months, if not weeks.

The organizations getting the most value have moved past optimizing individual projects. They're building the organizational capability to keep deploying AI as conditions change: evaluating new models against existing baselines, redesigning processes as model capabilities expand, and extending into new use cases as institutional confidence grows.

The organizations moving toward enterprise-wide impact have stopped treating AI deployment as a project with an end date and started treating it as an ongoing initiative with dedicated governance, metrics, and program management.

Before you move forward	
For the cross-functional team	**For the CIO or business leader**
Work out	**Assign**
Which workflows are candidates for full automation, and which should remain human-assisted?	Who is the human supervisor, exception handler, or system improver at scale?
What error modes exist at current volume, and what is the consequence of each at full volume?	Who owns the process for evaluating and adopting new model capabilities?
How do you measure success when the human is not in the loop?	Who owns ongoing AI program management as a function?

How production-ready organizations operate

Production-ready organizations define success criteria before the pilot begins. They classify data and resolve security and compliance requirements before setting the architecture. They commit to an infrastructure path and treat data readiness as a precondition, running the audit before locking the timeline. And they engage the real organizational stakeholders at the start, invest in workforce readiness, build governance that scales with the deployment, and treat scaling as an ongoing capability rather than a milestone.

The patterns of successful deployments we have documented throughout this guide reflect how enterprises can invest in the conditions that make new technology stick: clear ownership, data infrastructure, change management, and governance.

CIOs and business leaders must make the ownership, infrastructure, and governance decisions that determine whether an AI initiative reaches production before the pilot begins. The window for making those decisions is narrower than most planning cycles account for and organizations must treat deployment readiness as a strategic priority.

The next page translates the considerations we've outlined into a practical transition blueprint for what to decide, when to decide it, and who needs to own each decision before the program advances.

An enterprise AI deployment blueprint

PRE-PILOT
Decide before the pilot begins

01
Strategy and ownership
Set success criteria, ROI thresholds, and go/no-go gates.
- Executive sponsor

02
Data and integration
Audit the data estate, assign source and pipeline owners.
- Data and IT change owners

03
Infrastructure
Align platform choices to use case requirements.
- CIO / platform lead

04
Security and trust
Classify data and clear security, regulatory, and access requirements.
- Security and compliance lead

PILOT
Validate, don't defer

No new owners or decsions
The pilot tests the decisions already made.

Front-load the decisions
Every decision deferred past the pilot creates engineering debt: parallel systems, integration patterns that never standardize, and a platform foundation that's always being renegotiated.

PREPRODUCTION
Prepare the organization

05
Org readiness
Sequence rollout, fund reskilling, and name champions.
- Exec sponsor / change lead

PRODUCTION
Govern by risk

06
Governance and risk
Define risk taxonomy, monitoring thresholds, and review authority.
- Risk and monitoring owner

SCALED DEPLOYMENT
Operate as a capability

07
Scale and evolution
Start simple, scale deliberately. Understand how the system fails before scaling it.
- AI program owner

Getting started

Deploy and scale Claude across the enterprise

Claude Enterprise Administrator Guide: A four-phase guide to deploying Claude across an organization, from technical setup through scaling adoption.

Scaling agentic coding across your organization: A rollout playbook for moving Claude Code from individual users to organization-wide adoption.

Claude Code and new admin controls for business plans: An overview of admin controls for Claude Code, including spend caps, usage analytics, and the Compliance API.

Best practices for getting started with Claude Cowork: How to scope and run multi-step tasks in Claude Cowork for non-engineering teams.

Get started with Claude Cowork: A help-center walkthrough for switching on and using Claude Cowork.

Making Claude Cowork ready for enterprise: An overview of Claude Cowork's enterprise controls, including role-based access and per-team budgets.

Build with Claude

Best practices for Claude Code: Anthropic's engineering guide to working effectively in Claude Code, from planning to test-driven workflows.

How Claude Code is used in practice: Anthropic research on how people use Claude Code across roles and tasks.

Building effective agents: Anthropic's primer on when to build an agent versus a simpler workflow, with reusable design patterns.

Effective context engineering for AI agents: Guidance on what to load into a model's context window and what to leave out as systems grow.

Writing effective tools for AI agents: How to design and document the tools an agent calls so it can use them reliably.

How we built our multi-agent research system: A behind-the-scenes account of the architecture, evaluation, and failure modes from Anthropic's own multi-agent system.

Contextual Retrieval: A preprocessing method that improves retrieval accuracy by attaching context to each chunk before indexing.

Introducing the Model Context Protocol: The open standard for connecting Claude to enterprise data sources and tools through one interface.

Define success criteria and build evaluations: Anthropic's documentation on writing measurable success criteria and building evals to test against them.

Our framework for developing safe and trustworthy agents: Anthropic's early framework for agent safety, built on principles like keeping humans in control and securing how agents act.

About Anthropic

Anthropic is an AI research and development company that creates reliable, interpretable, and steerable AI systems. Anthropic's flagship product is Claude, a large language model trusted by millions of users worldwide. Learn more about Anthropic and Claude at anthropic.com.

About Accenture

Accenture helps the world's leading enterprises reinvent by building their digital core and unleashing the power of AI to create value at speed for organizations across industries. Our strategy is to be the reinvention partner of choice for our clients and lead in the safe, widespread adoption of AI, and to be the most client-focused, AI-enabled, great place to work in the world. We bring together the talent of our approximately 799,000 people with proprietary assets and platforms, deep process and industry expertise, and leading ecosystem relationships to deliver end-to-end solutions and measurable outcomes at scale. Through our Reinvention Services, we offer broad expertise across Cybersecurity, Digital Core, Finance, Industry and Enterprise, Song, Supply Chain and Engineering, and Talent, with advanced capabilities in AI and Data, Industry and Process, and Technology.

ANTHROP\C | accenture