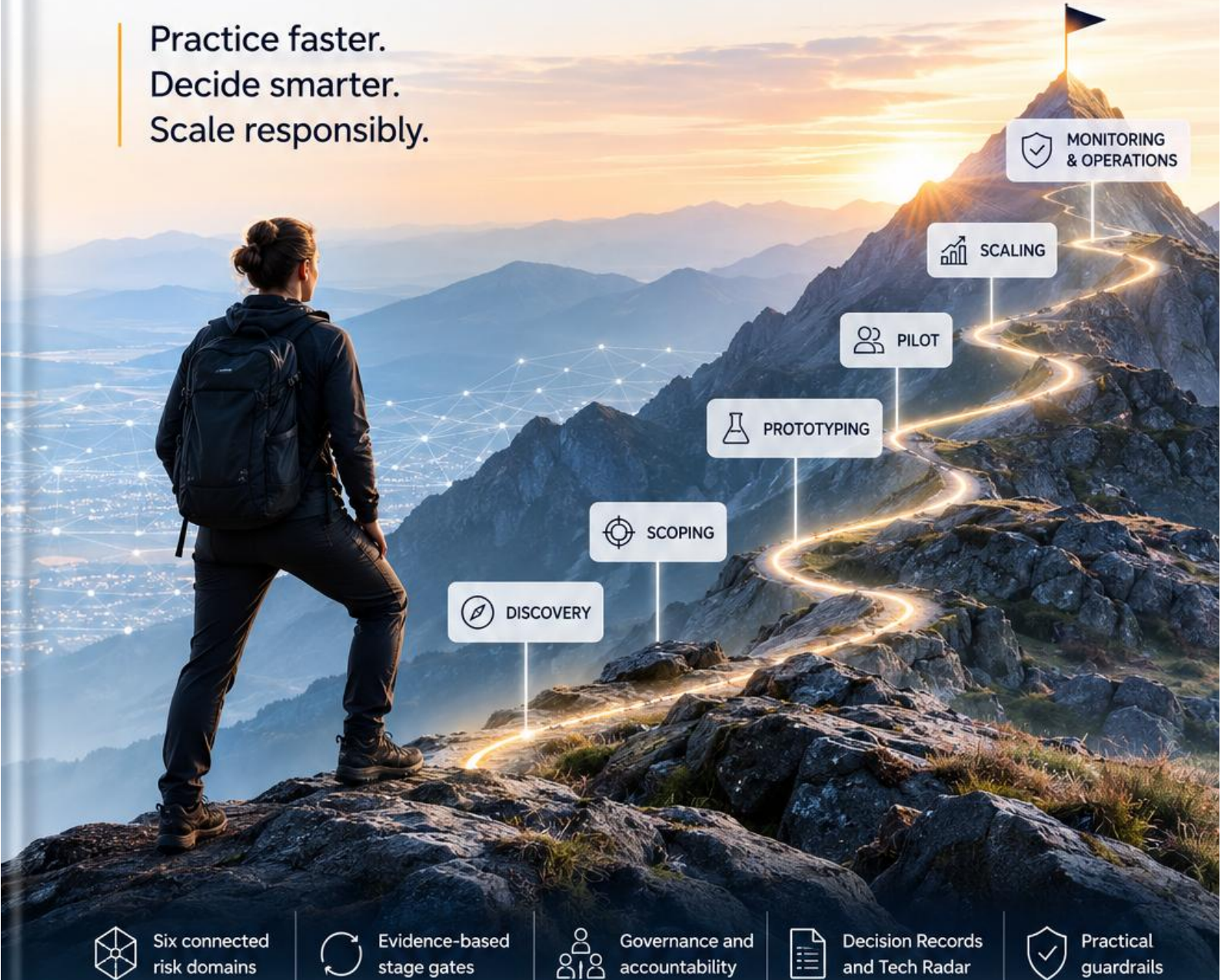
A PRACTICAL HANDBOOK

How to Manage Risks in AI Projects

A Practical Handbook for Combining Agile Learning with Structured Risk Decisions

Practice faster.
Decide smarter.
Scale responsibly.



- Six connected risk domains
- Evidence-based stage gates
- Governance and accountability
- Decision Records and Tech Radar
- Practical guardrails

DAVID THEIL
DIGITALISIERUNGSCOACH

DigitalisierungsCoach
Practical. Strategic. Human.

How to Manage Risks in AI Projects

A Practical Handbook for Combining Agile Learning with Structured Risk Decisions

MOVE FAST — BUT RESPONSIBLY

DAVID THEIL
AI-Strategist and Organizational Development

Version 1.0 | 16 July 2026

Innovation × Trust × Compliance = Sustainable AI Success

Publication Information

Author
David Theil

Organization / publication platform
DigitalisierungsCoach

Publication date
16 July 2026

Version
1.0 — First handbook edition

Language
English

Online profiles
DigitalisierungsCoach | LinkedIn | Medium

Recommended citation

Theil, David. How to Manage Risks in AI Projects: A Practical Handbook for Combining Agile Learning with Structured Risk Decisions. Version 1.0. DigitalisierungsCoach, 2026.

Copyright notice

Version note

This handbook reflects the sources and regulatory materials available at the publication date. AI technologies, regulatory guidance, standards, and provider conditions continue to evolve. Readers should verify the latest official versions before applying the handbook in a specific context.

Executive Summary

Artificial intelligence projects cannot be managed as conventional IT projects with an additional technology component. Traditional concerns such as architecture, security, budget, scope, integration, and organizational readiness remain important, but AI introduces further uncertainty around data quality, probabilistic model behavior, third-party providers, drift, human interaction, adoption, ethics, regulation, and trust.

At the same time, organizations cannot eliminate all uncertainty before they begin. AI initiatives need experimentation. Teams must test assumptions, work with limited prototypes, involve users, and learn from evidence generated under realistic conditions.

Agile iterations manage learning. Structured stage gates manage organizational exposure.

This handbook presents a practical operating model that combines both mechanisms. Inside each phase, teams iterate, learn, and adapt. At defined decision points, authorized leaders assess whether the evidence justifies additional investment, broader access to data, more users, deeper integration, increased autonomy, or wider operational use.

The core model

Element	Purpose
Expanded risk profile	AI risk extends beyond technical and project risk to value, viability, data, models, operations, compliance, ethics, and trust.
Six connected risk domains	Risks are examined as one socio-technical system rather than as isolated departmental checklists.
Continuous lifecycle management	The risk assessment changes when the use case, data, model, provider, users, controls, or consequences change.
Iterate, learn, adapt, decide	Small, bounded experiments convert uncertainty into evidence and evidence into an explicit next step.
Evidence-driven gates	A gate determines whether the next level of organizational exposure is justified.
Explicit accountability	Risk owners, control owners, evidence owners, acceptance authorities, and stop authorities are distinguished.
Traceable decisions	Decision Records preserve context, evidence, residual risks, conditions, and review triggers.
Practical guardrails	Teams receive clear boundaries for tools, data, human oversight, traceability, monitoring, and escalation.
Monitoring and reassessment	Operational evidence challenges earlier assumptions and reopens decisions when conditions change.

The decision principle

The purpose of AI risk management is not to eliminate every risk before an initiative begins. It is to make uncertainty visible, test the most consequential assumptions, limit the possible impact of failure, reduce risks where possible, and consciously accept the residual exposure.

Risk management should not prevent experimentation. It should prevent organizations from scaling assumptions that have not yet earned their trust.

The handbook therefore treats every approval as conditional and reviewable. A decision should specify what is permitted, under which conditions, with which controls, for which users and data, who accepts the remaining risk, and which events require reassessment.

Who This Handbook Is For

This handbook is written for people who design, sponsor, build, govern, deploy, or operate AI-enabled products and services. It is intended to create a shared language across business, technology, risk, and organizational functions.

Primary audience	How the handbook helps
Executives and sponsors	Use the handbook to define risk appetite, evaluate investment decisions, and understand which exposure is being accepted.
Product, project, and AI initiative owners	Use it to structure the complete risk picture, define experiments, prepare evidence, and coordinate decisions.
Agile coaches and change practitioners	Use it to connect iterative learning with governance, adoption, human factors, and organizational change.
Data scientists and software engineers	Use it to connect technical evaluation with business context, operating controls, traceability, and lifecycle monitoring.
Security, privacy, legal, and compliance specialists	Use it to shape early experiments, define non-negotiable boundaries, and identify when formal review is required.
Service owners and operations teams	Use it to define monitoring, incident response, provider change management, fallback options, and retirement criteria.
HR, employee representatives, UX, and domain experts	Use it to assess human impact, fairness, workflow fit, calibrated trust, and meaningful oversight.

Scope and Boundaries

The handbook focuses on the organizational management of AI initiatives from discovery through scoping, prototyping, piloting, rollout, and ongoing operations. It applies to AI systems developed internally, procured from vendors, embedded in third-party products, or assembled from external models, platforms, data sources, and services.

Included in scope

- identifying and structuring AI-specific and conventional risks across six connected domains;
- designing responsible experiments, learning cycles, and evidence-driven stage-gate decisions;
- defining evidence requirements for each lifecycle phase;
- assigning risk ownership, control ownership, decision authority, and stop authority;
- preserving decisions through Decision Records, organizational technology guidance, and practical guardrails;
- monitoring changing conditions and diagnosing common AI risk-management anti-patterns;

Not included as a substitute for specialist work

This handbook does not provide a complete legal interpretation of the EU AI Act or other legislation, a conformity-assessment method, a certification implementation guide for ISO/IEC 42001, a detailed cybersecurity control catalogue, or a model-evaluation standard for every technical domain.

- Legal classification, obligations, security architecture, and controls must be assessed by qualified legal, compliance, and security professionals.
- Sector-specific requirements may impose additional duties not covered here.
- Model thresholds and evaluation methods must be tailored to the intended use, affected people, and consequences.

Central definition: organizational exposure

In this handbook, organizational exposure means the combined level of reach, autonomy, data sensitivity, potential consequence, reversibility, operational dependency, and resource commitment introduced by an AI initiative.

The strength of governance, evidence, controls, and oversight should increase with organizational exposure.

How to Use This Handbook

The chapters form one connected operating model and can be read sequentially. The handbook can also be used as a reference during initiative design, gate preparation, governance reviews, or operational reassessment.

Situation	Recommended starting point
You are evaluating a new AI idea	Start with the expanded risk profile, the six risk domains, and the discovery and scoping gates.
You are preparing a prototype or pilot	Use Iterate–Learn–Adapt–Decide, the lifecycle chapter, and the relevant evidence requirements.
You are designing AI governance	Use the governance, Decision Records, Tech Radar, guardrails, and anti-pattern sections.
You are reviewing an operational AI system	Use monitoring triggers, operational evidence, residual-risk ownership, and renewal decisions.
You need to challenge a weak approval process	Use the anti-pattern diagnostic and ask what evidence, conditions, owners, and stop triggers are missing.

Use the model as a decision system, not as a checklist

The framework should be tailored to the organization, use case, potential impact, and available evidence. Not every initiative requires the same level of documentation or specialist review. A low-exposure experiment should remain lightweight; a consequential system requires stronger evidence, clearer accountability, and continuous monitoring.

Recommended working rhythm

1. Define the intended use, excluded uses, users, data, and operating boundaries.
2. Identify the most consequential and uncertain assumptions across the six risk domains.
3. Design the smallest responsible experiment that can produce useful evidence.
4. Evaluate both the AI component and the complete socio-technical work system.
5. Adapt the model, data, process, controls, scope, or level of autonomy.
6. Make an explicit Go, Rework, Restrict, Pause, or Kill decision.
7. Record the decision, residual risks, conditions, owners, and review triggers.
8. Monitor the system and reopen the decision when conditions change.

Contents

Introduction

AI projects are often treated like conventional IT projects with a new technology at their center.

A project team is formed, a scope is defined, costs are estimated, and technical risks are documented. Legal, security, and data-protection specialists may be involved before the solution is released.

This approach remains necessary, but it is no longer sufficient. The *NIST* AI Risk Management Framework notes that AI systems introduce risks that are not comprehensively addressed by traditional software risk frameworks, including risks related to changing training conditions, outdated datasets, third-party models, scale, and use outside the originally intended context.

AI projects introduce uncertainties that cannot be managed through conventional project planning and technical quality assurance alone. The behavior of an AI solution depends not only on its code, but also on data, models, providers, user interactions, business context, and the environment in which it operates.

Some of these factors can change even when the project team does not change the solution itself.

A model provider may release a new version. Input data may become outdated. User behavior may differ from what the team expected. A system that performs well in a controlled prototype may produce different results when it is exposed to real users, real workloads, and real business decisions.

At the same time, organizations cannot eliminate all uncertainty before they begin.

AI projects need experimentation. Teams have to test assumptions, work with prototypes, involve users, and learn from real results. Long planning cycles and extensive upfront specifications cannot answer all the questions that emerge during the development of an AI solution.

This creates a fundamental challenge:

AI projects need enough freedom to learn quickly, but enough structure to prevent uncontrolled risk.

Too much control can slow down experimentation and prevent innovation.

Too little control can expose the organization, its employees, and its customers to technical, economic, legal, ethical, and reputational harm.

The solution is not to choose between agility and governance.

The solution is to combine them.

Agile practices help teams expose uncertainty, test assumptions, and adapt the solution through short learning cycles. Structured risk decisions help the organization determine whether the available evidence justifies further investment, broader usage, access to more sensitive data, or increased operational responsibility.

This handbook introduces a practical model that combines both approaches.

It explains:

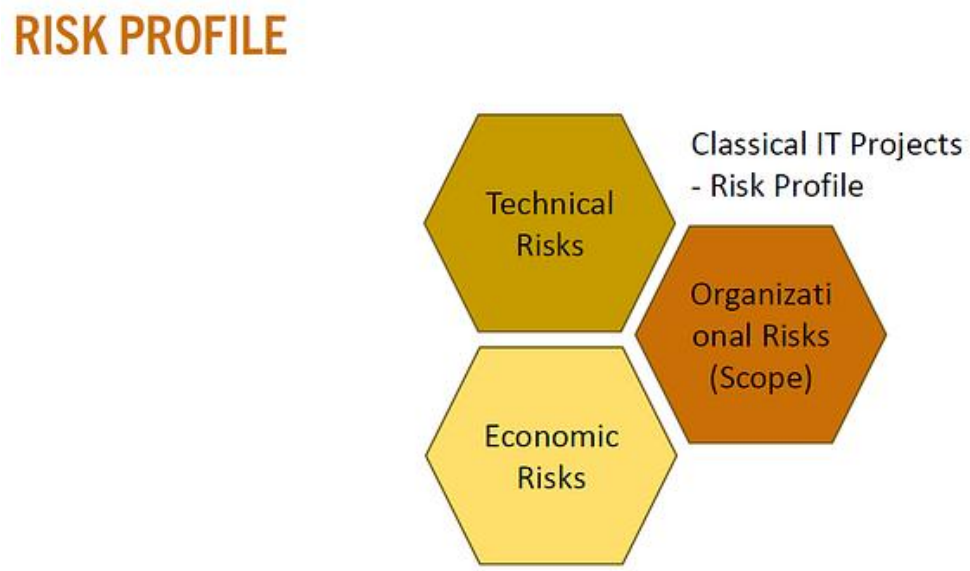
- why AI projects require a different risk-management approach,
- how their risk profile differs from conventional IT projects,
- which risk domains need to be managed,
- how agile learning can be combined with stage-gate decisions,
- what evidence is needed before an initiative moves forward,
- and how governance, traceability, and practical guardrails support responsible innovation.

The goal is not to remove every risk before an AI initiative begins.

The goal is to make risks visible early, reduce them through evidence, and increase the organization's exposure only when the results justify it.

Why AI Projects Need a Different Risk Management Approach

Conventional IT projects already involve considerable uncertainty.



Risk Profile of Conventional IT Projects

Organizations have to manage technical feasibility, system integration, security, scope, dependencies, costs, schedules, and organizational change. These risks do not disappear when AI is introduced. They remain part of every AI initiative.

However, AI projects add another layer of uncertainty.

In traditional software development, developers define explicit rules that determine how the system should behave. Under the same conditions, the software is generally expected to produce the same result. When something goes wrong, the cause can often be traced to a defect in the code, a misunderstood requirement, incorrect data, or a failed integration.

AI systems are different. Their behavior is influenced by probabilities, training data, model characteristics, prompts, context, and user interactions. Their outputs may be plausible without being correct. They may perform well for common cases but fail in situations that were not sufficiently represented during testing.

AI solutions can also change after they have been deployed.

The data may drift. The model provider may update the underlying model. New user groups may use the system differently. The solution may be expanded to processes with higher consequences. A decision-support tool may gradually become an automated decision system without the organization explicitly acknowledging that change.

This means that risk management cannot be limited to a technical review before rollout.

It must accompany the entire lifecycle of the initiative.

The organization needs to understand not only whether the solution can be built, but also:

- whether it solves a relevant problem,
- whether its business case remains viable,
- whether the underlying data can be trusted,
- whether the model behaves reliably,
- whether the solution can be operated safely,
- and whether its use remains fair, compliant, and trustworthy.

This is the expanded risk profile of an AI project.

Before defining how these risks should be managed, we first need to understand how this risk profile differs from that of a conventional IT project.

AI Projects Have an Expanded Risk Profile

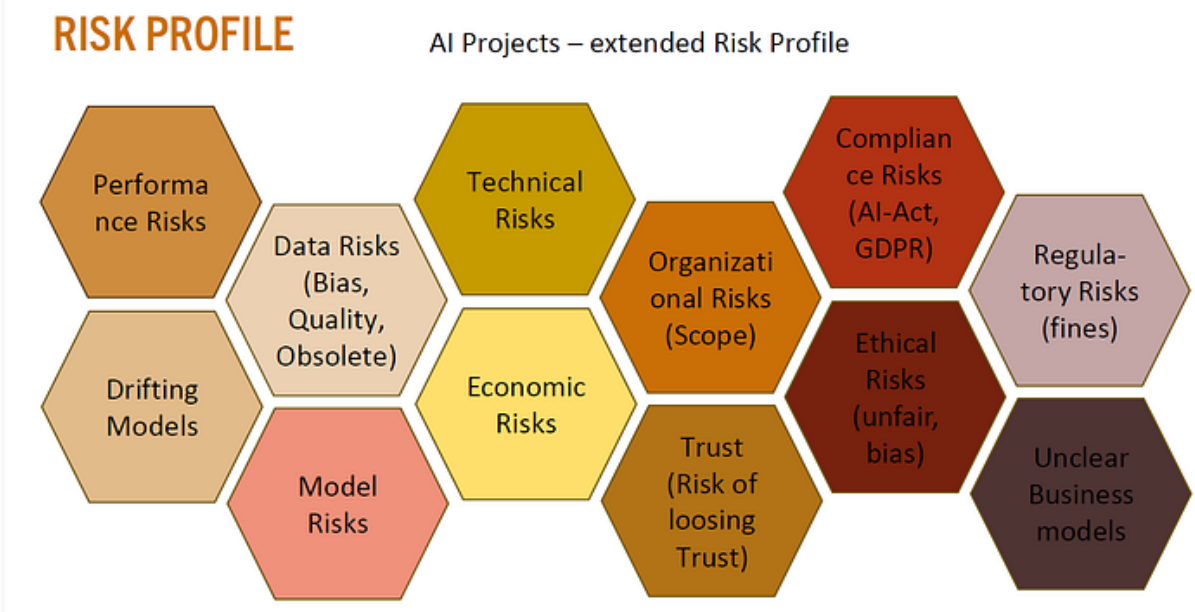
Conventional IT projects already involve substantial risks.

A project may exceed its budget, miss its delivery date, fail to integrate with existing systems, or produce a solution that does not satisfy the original requirements. Organizations therefore monitor technical feasibility, scope, costs, dependencies, security, and organizational readiness.

All these risks remain relevant in AI projects.

The difference is that AI adds further sources of uncertainty that are not fully controlled by the project team and cannot always be addressed through conventional software testing.

The risk profile does not simply shift. It expands.



Example: Extended Risk Profile of AI Projects

A conventional IT project typically focuses on three broad areas:

- **Technical risks:** Can the solution be built, integrated, secured, and operated?
- **Organizational risks**: Is the scope clear? Are responsibilities, skills, and dependencies properly managed?
- **Economic risks:** Can the solution be delivered within the available time and budget, and will the expected value justify the investment?

AI projects have to manage these same risks, but they also introduce uncertainty around the data, the model, the behavior of users, external providers, regulation, ethics, and trust.

This expanded risk profile is caused by several characteristics of AI systems.

AI Output Is Not Fully Defined by the Code

In most conventional business software, developers define explicit rules for how the system should respond.

A calculation follows a formula. A workflow follows predefined conditions. A validation rule accepts or rejects an input according to criteria written into the software.

AI systems do not operate exclusively through such explicit rules.

Their output is also shaped by training data, statistical relationships, model architecture, prompts, context, parameters, and user input. This means that the project team cannot describe every possible result in advance.

The output may be plausible but incorrect.

It may be useful in one situation and misleading in another. It may perform reliably for common cases but fail when confronted with unusual inputs or contexts that were not sufficiently represented during testing.

Testing an AI system is therefore not only about finding defects in the implementation. It is also about understanding the boundaries within which the system can be trusted.

The Data Becomes Part of the System's Behavior

Data quality is important in every IT project, but it has an even more fundamental role in AI projects.

The data does not merely provide information to the system. It influences how the system behaves.

Incomplete, outdated, biased, incorrectly classified, or improperly licensed data can affect the quality and fairness of the output. In some cases, the project team may not even have full visibility into the data used to train an external model.

This creates questions that conventional testing alone cannot answer:

- Is the available data representative of the real operating environment?
- Which groups or situations may be underrepresented?
- Is the data sufficiently current?
- Is the organization legally permitted to use it?
- Can the origin and quality of the data be traced?
- What happens when the data changes over time?

Poor data does not only create isolated errors. It may create recurring patterns of incorrect or unfair behavior.

Models and Providers Can Change

AI projects increasingly depend on third-party models, APIs, platforms, and data sources. NIST therefore recommends continuous monitoring of third-party AI systems, explicit supplier-risk assessments, fallback strategies, and contractual provisions that address system changes over time, including fine-tuning, drift, and performance decay.

These dependencies accelerate development, but they also reduce the organization's direct control over important parts of the solution.

A provider may:

- release a new model version,
- change model behavior,
- discontinue an existing model,
- modify pricing,
- introduce new usage limits,
- change licensing conditions,
- or alter the way customer data is processed.

The project may therefore change even when the internal team has not modified its own code.

An AI solution that is economically viable today may become too expensive when usage increases or pricing changes. A model that passed the original tests may behave differently after an external update. A solution that depends deeply on one provider may be difficult or costly to migrate.

Provider dependency is therefore not only a technical architecture question. It is also a strategic, operational, and economic risk.

Users Influence the Risk Profile

The behavior of an AI system cannot be assessed independently from the people who use it.

Users may rely too strongly on its recommendations. They may use it for purposes that were never intended. They may share confidential information, misunderstand the limitations of the model, or stop checking outputs once the system has earned their trust.

The opposite can also happen.

Employees may reject a useful solution because they do not understand it, do not trust it, or fear that it will eventually replace them. A technically successful system can therefore fail because of low adoption, missing skills, inappropriate usage, or active resistance.

Human behavior is not merely a change-management concern added after the technical work is complete. It is part of the AI system's risk profile.

It is part of the risk profile of the AI system itself.

Small Errors Can Create Large Consequences at Scale

During prototyping, an incorrect AI output may affect only a few test cases.

During a pilot, it may influence a limited number of users or decisions.

After rollout, the same weakness may be repeated thousands of times across departments, customer interactions, or automated processes.

Scaling therefore does not only increase the potential value of an AI solution. It also increases the potential impact of its weaknesses.

This is especially critical when AI is used to:

- recommend actions,
- prioritize cases,
- generate customer communication,
- support personnel decisions,
- assess individuals,
- produce software code,
- or automate parts of regulated processes.

The higher the reach, autonomy, and consequence of the system, the stronger the required controls must become.

Trust Is Both an Outcome and a Risk

Trust plays a special role in AI projects.

Employees and customers need enough trust to use the system. At the same time, excessive trust can be dangerous when people stop questioning the output.

Organizations therefore need to establish **appropriate trust.**

Users should understand:

- what the system can do,
- where its limitations are,
- when human judgment is required,
- how outputs can be challenged,
- and who remains accountable for the final decision.

Trust takes time to build, but a serious incident can destroy it quickly.

An unfair recommendation, a privacy violation, a public hallucination, or an unexplained automated decision may affect not only one AI initiative. It may reduce confidence in the organization's broader AI strategy.

Risk management is therefore also a trust-building mechanism.

Trust should not mean unquestioned reliance. Users need enough information, competence, and authority to interpret AI output, recognize its limitations, intervene, and stop the system when necessary. This principle is also reflected in the human-oversight requirements of the EU-AI Act for high-risk AI systems.

It demonstrates that the organization does not treat AI as a technological experiment without consequences, but as a system that requires transparency, accountability, and continuous oversight.

From Additional Risks to a Structured Risk Model

The expanded risk profile of AI projects is not a reason to avoid AI.

It is a reason to manage AI initiatives differently.

Organizations need to preserve the ability to experiment and learn, while recognizing that technical performance is only one part of a successful and responsible AI solution.

A model may be technically impressive but deliver no relevant value.

It may deliver value but have an unsustainable business case.

It may be economically attractive but rely on poor data.

It may perform well but be impossible to explain, monitor, or maintain.

And it may satisfy all technical requirements while still creating ethical, regulatory, or trust-related problems.

To manage this expanded profile systematically, we can group the risks of AI projects into six connected domains.

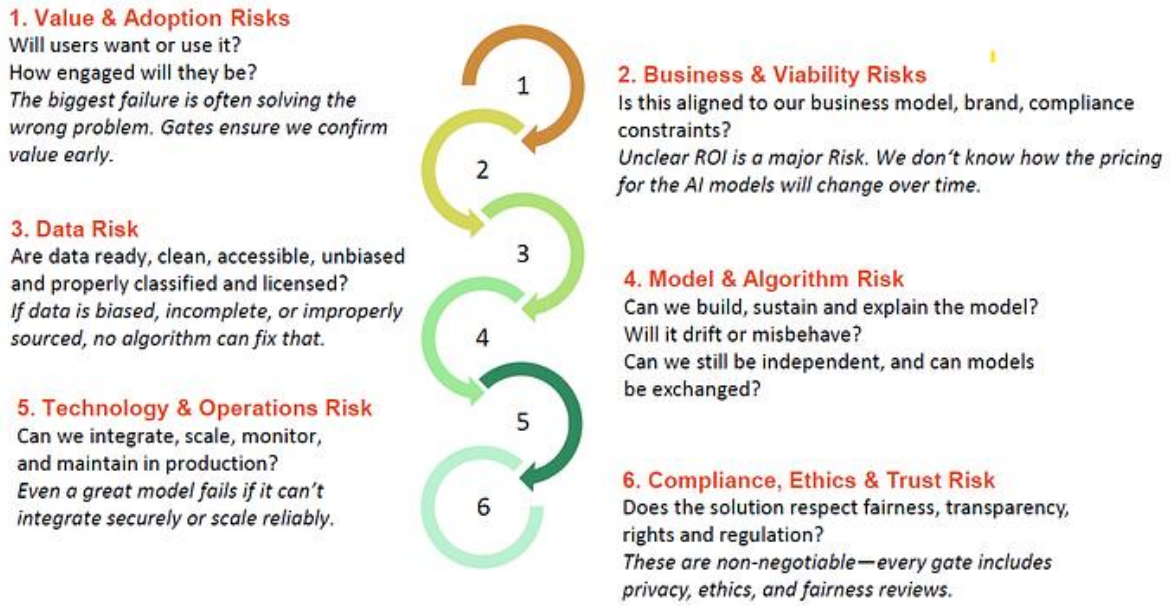
The Six Risk Domains of AI Projects

An expanded risk profile can quickly become an unmanageable collection of individual concerns.

Data quality, model performance, security, compliance, user adoption, costs, ethics, and operational readiness may all appear on different checklists owned by different departments. Without a shared structure, risks are easily assessed in isolation, duplicated across reviews, or overlooked because everybody assumes that somebody else is responsible.

To make the expanded risk profile manageable, we can organize the risks of an AI project into six connected domains:

1. Value and Adoption Risks
2. Business and Viability Risks
3. Data Risks
4. Model and Algorithm Risks
5. Technology and Operations Risks
6. Compliance, Ethics, and Trust Risks



The Six Risk Domains of AI Projects

These domains are not independent categories.

A problem that begins in one domain can quickly create risks in several others. Poor data quality can reduce model performance. Incorrect model output can lead to harmful business decisions. Harmful decisions can create legal problems and cause employees or customers to lose trust.

The purpose of the six-domain model is therefore not to assign every risk to exactly one box.

Its purpose is to make sure that the project team examines the AI initiative from all relevant perspectives and understands how the risks influence one another.

1. Value and Adoption Risks

The first question should not be whether the AI solution can be built. It should be whether it should be built at all.

Many AI initiatives begin with the technology rather than the problem. A team discovers a new model, platform, or AI capability and starts looking for possible applications. This can produce impressive prototypes, but it does not necessarily produce valuable solutions.

The fundamental value risk is that the initiative solves the wrong problem — or solves a real problem that is not important enough to justify the required investment and organizational change.

Ask:

- Does the initiative address a relevant business or user problem?
- Is AI actually required to solve it?
- Is the proposed solution better than the current process or a simpler non-AI alternative?
- Will users adopt the solution in their daily work?
- Does it improve a measurable outcome?
- Are users able to recognize when they should and should not rely on it?

Adoption is part of this risk domain because value only emerges when the solution is used appropriately.

A technically successful system may still fail when employees do not trust it, do not understand it, or see it as an additional burden. The opposite can also become a risk: users may trust the system too much and stop critically reviewing its output.

The objective is therefore not maximum trust or maximum usage.

It is **appropriate usage and calibrated trust.**

2. Business and Viability Risks

An AI solution can deliver value to individual users and still fail as a sustainable business initiative.

The business and viability domain examines whether the solution remains economically and strategically reasonable beyond the prototype.

AI projects frequently depend on external model providers, APIs, cloud services, licenses, and specialized infrastructure. These dependencies may introduce costs and conditions that are difficult to forecast during the early stages of the project.

Ask:

- Is the initiative aligned with the organization’s objectives and business model?
- Is the expected value greater than the cost of development, rollout, operation, and monitoring?
- What happens when usage increases significantly?
- How sensitive is the business case to changing model or infrastructure costs?
- Are licensing and contractual conditions sufficiently clear?
- Can the organization switch providers or models if necessary?
- Who owns the long-term budget after the project becomes an operational service?

An unclear business model is not a problem that should be postponed until scaling. It is a risk that should be tested from the beginning.

A prototype may be inexpensive because it is used by only a few people. The same solution may become economically unattractive when it processes thousands of requests, requires continuous human review, or depends on expensive infrastructure.

Technical feasibility does not automatically create business viability.

3. Data Risks

Data is not merely an input to an AI system. It shapes the behavior and limitations of the solution.

Poor-quality, incomplete, outdated, biased, or improperly sourced data can result in unreliable or unfair output. In some projects, the organization may also lack sufficient knowledge about the data used to train an external model.

Relevant questions include:

- Is sufficient data available for the intended use case?
- Is the data accurate, representative, and current?
- Which users, situations, languages, or edge cases may be underrepresented?
- Is the organization permitted to use the data for this purpose?
- Is personal or confidential information adequately protected?
- Can the origin and transformation of important data be traced?
- How will changes in data quality be detected?
- Who owns the data and is responsible for maintaining it?

Data risks exist throughout the lifecycle. During discovery, the organization may discover that the required data does not exist. During prototyping, the team may find quality or representation problems. During operations, the data may gradually become outdated or diverge from the conditions under which the system was originally tested.

The familiar principle still applies:

Garbage in, garbage out.

However, in AI projects, the consequences can be harder to detect because the output may still appear convincing.

4. Model and Algorithm Risks

Model and algorithm risks concern whether the AI component behaves reliably enough for its intended purpose.

Unlike conventional software functions, AI models may produce probabilistic, inconsistent, or unexpected results. Their output may appear plausible while being incomplete or incorrect.

The relevant questions are therefore broader than whether the model achieves a single accuracy target.

Ask:

- Does the model perform sufficiently well for the intended use case?fv
- Which types of errors occur?

- What are the consequences of false positives and false negatives?
- How does the model behave in unusual or previously unseen situations?
- Can users understand the basis and limitations of its output?
- Can important results be reproduced and traced?
- How will drift or performance degradation be detected?
- What happens when the provider updates or replaces the model?
- Can the model be exchanged without redesigning the entire solution?

Model quality must always be evaluated in context.

An error rate that is acceptable for generating internal meeting summaries may be unacceptable for decisions affecting employment, legal obligations, customer rights, or safety.

The question is therefore not:

“Is the model accurate?”

The better question is:

“Is the model sufficiently reliable for this specific purpose, under these specific conditions, with these specific safeguards?”

5. Technology and Operations Risks

A working model is not yet a working operational system. The AI solution must also be integrated, secured, scaled, monitored, maintained, and supported. This is where many promising prototypes fail.

Technology and operations risks include:

- integration with existing systems,
- access control and security,
- infrastructure reliability,
- scalability,
- response times,
- logging and observability,
- monitoring of costs and performance,
- incident management,
- maintenance responsibilities,
- and dependency on external services.

Ask:

- Can the solution be integrated securely into the existing environment?
- Can it handle the expected workload?
- Are critical inputs and outputs logged appropriately?
- Can incidents and unexpected behavior be detected?

- Is there a fallback when the model or provider is unavailable?
- Who operates and supports the solution?
- Who is authorized to change the model, prompts, data, or configuration?
- Under which conditions will the system be restricted, rolled back, or stopped?

A prototype is usually optimized for learning. An operational system must be optimized for reliability, control, and maintainability. The transition between the two should never be treated as a simple technical deployment.

6. Compliance, Ethics, and Trust Risks

The final domain covers whether the AI solution can be used lawfully, fairly, transparently, and responsibly.

Compliance and ethics should not be treated as reviews that happen only shortly before launch. They influence the design of the system, the permitted data, the required documentation, the role of human oversight, and sometimes whether the use case should be pursued at all.

Ask:

- Which laws, regulations, contracts, and internal policies apply?
- Could the solution discriminate against individuals or groups?
- Are affected users informed when AI is involved?
- Can important outputs and decisions be explained?
- Is meaningful human oversight possible?
- Can a person challenge, correct, or stop an AI-supported decision?
- Is accountability clearly assigned?
- Could the solution damage employee, customer, or public trust?

Trust is particularly fragile.

A serious privacy incident, discriminatory outcome, unexplained decision, or confidently presented false statement may damage more than one initiative. It can undermine confidence in the organization's entire AI strategy.

Risk management is therefore not merely a mechanism for avoiding fines or incidents.

It is also a mechanism for building and maintaining trust through transparency, accountability, fairness, security, and responsible human oversight.

The Six Domains Form One Risk System

The six domains provide a shared structure, but they should never become six separate checklists managed independently by six departments.

Consider a recruiting assistant trained on incomplete historical data:

- The **data risk** is that the historical information is not representative.
- The **model risk** is that the system reproduces or amplifies hidden patterns.
- The **compliance and ethics risk** is that applicants may be treated unfairly.
- The **trust risk** is that employees and applicants lose confidence in the process.
- The **adoption risk** is that recruiters reject the system — or rely on it without sufficient scrutiny.
- The **business risk** is that the initiative creates reputational and financial damage instead of operational value.

This illustrates why AI risk management must be cross-functional. No single project manager, data scientist, security specialist, or legal expert can assess the entire risk profile alone. Each role sees only part of the system.

The six domains create a common language through which these perspectives can be combined. However, identifying the relevant risk domains is only the first step.

The risk profile will change as the initiative moves from an initial idea to a prototype, from a prototype to a real-world pilot, and finally into broad operational use. A risk assessment performed once at the beginning of the project will therefore become outdated long before the system reaches production.

Why a One-Time Risk Assessment Is Not Enough

The six risk domains provide a structure for identifying the risks of an AI project.

However, they only provide a snapshot.

A risk assessment performed during the scoping phase describes what the organization knows about the initiative at that moment. It is based on the intended use case, the available data, the selected model, the expected users, and the planned operating environment.

All of these factors can change.

As the initiative moves from an idea to a prototype, from a prototype to a pilot, and from a pilot into broad operational use, the organization learns more about the solution. New risks become visible, previous assumptions are invalidated, and controls that seemed sufficient in a small test environment may no longer be effective at scale.

Risk management must therefore evolve together with the AI system.

A Risk Register Is Only as Current as Its Assumptions

Many projects create a risk register during planning.

The team identifies potential problems, estimates their probability and impact, defines mitigation measures, and assigns owners. The register is then reviewed periodically or presented at steering meetings.

This approach can create the appearance of control. But it becomes dangerous when the original assessment is treated as a stable description of the project.

Every risk entry is based on assumptions:

- how the system will be used,
- which data it will process,
- how the model will behave,
- how many users will interact with it,
- which decisions it will influence,
- which provider and infrastructure it will depend on,
- and which safeguards will remain effective.

When these assumptions change, the risk assessment must change as well.

A risk marked as acceptable during prototyping may become unacceptable when the system is connected to production data. A control that works for ten trained pilot users may fail when thousands of employees gain access. A low-impact error in an internal assistant may become a serious risk when the same technology is used for customer communication or decisions affecting individuals.

The relevant question is therefore not:

“Have we completed the risk assessment?”

It is:

“Is our current understanding of the risks still valid under the conditions in which the system is now being used?”

The Risk Profile Changes Even When the Code Does Not

In conventional software projects, a significant change in risk is often connected to a software release, architectural modification, or change in requirements.

AI systems can change in more subtle ways.

The project team may leave its own application code untouched while:

- the model provider introduces a new model version,
- the behavior of an external API changes,
- the distribution or quality of input data shifts,
- users develop new ways of prompting or interacting with the system,
- the solution is applied to a new process,
- the volume of usage increases,
- or the threat landscape changes.

NIST therefore treats AI risk management as a lifecycle activity rather than a single evaluation step. Its AI Risk Management Framework organizes risk work through the connected functions **Govern, Map, Measure, and Manage**. These functions are intended to support recurring risk-related activities throughout the design, development, deployment, use, and evaluation of AI systems.

The implication is important:

A system does not remain safe simply because it passed an earlier review.

Its risk profile depends on the current combination of model, data, users, context, controls, and consequences.

New Risks Become Visible in Different Phases

Not every risk can be assessed with the same level of confidence at the beginning of an initiative.

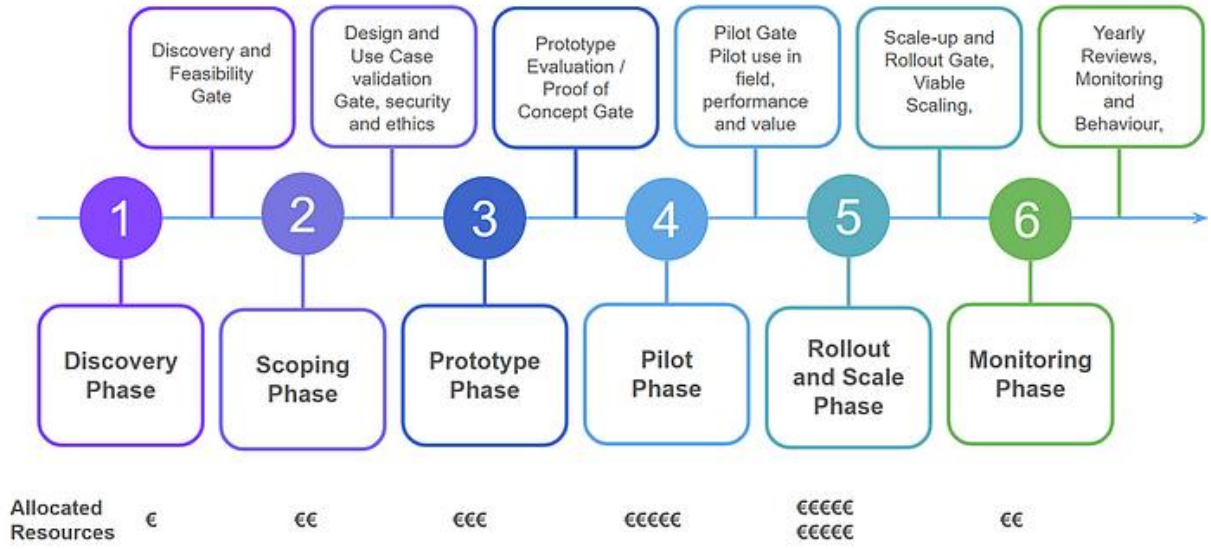
During the **discovery phase**, the team may mainly understand the intended problem and expected business value. At that point, detailed statements about model reliability or operational scalability may still be based on assumptions.

During **prototyping**, the team gains evidence about data availability, technical feasibility, and initial model performance. However, the system is usually tested with a limited dataset and in a controlled environment.

During the **pilot** phase, risks related to real users, behavior, adoption, integration, security, and human oversight become more visible.

During **scaling**, small weaknesses may become significant because they are repeated across more users, decisions, processes, and data.

During **operations**, the organization must monitor whether the model, data, system performance, costs, controls, and usage patterns continue to remain within acceptable boundaries.



AI Project Phases and Stage Gates, in contrast to the allocated resources

The purpose of lifecycle risk management is not to predict every possible problem during discovery. It is to make sure that the organization asks the right risk questions as soon as reliable evidence becomes available.

Controls Can Lose Their Effectiveness

Risk management is not complete when a mitigation measure has been defined. The organization also needs evidence that the measure works. Consider a project team that identifies hallucinations as a model risk and introduces human review as a safeguard. During a limited pilot, every generated response may be checked carefully by a small group of motivated users.

After rollout, the situation may change:

- users are under greater time pressure,
- the number of outputs increases,
- reviewers begin to trust the system,
- the review becomes a routine confirmation step,
- or responsibility for checking the output becomes unclear.

The documented control still exists, but its practical effectiveness has declined.

The same problem may occur with:

- data-quality checks that are no longer performed,
- monitoring thresholds that were never updated,
- fallback procedures that have not been tested,
- training that does not reach new users,
- or security reviews that do not cover later integrations.

NIST's AI RMF Playbook therefore recommends regular monitoring for performance degradation, unexpected behavior, near misses, attacks, and changing system impacts. It also recommends

continuously monitoring the performance and trustworthiness of pre-trained models and third-party AI components.

Risk controls must be monitored just like the AI system itself.

Continuous Risk Management Is Also a Regulatory Principle

The EU-AI Act reflects this lifecycle logic for high-risk AI systems.

Article 9 requires a risk management system for high-risk AI systems and defines it as a continuous, iterative process that is planned and performed throughout the entire lifecycle. The process requires regular systematic review and updating, as well as the identification, analysis, evaluation, and treatment of risks.

This legal obligation applies specifically to systems classified as high-risk under the AI Act. It should therefore not be presented as a legal requirement for every AI initiative.

However, the underlying management principle is useful far beyond the legally defined high-risk category:

When the system, its context, or its potential impact changes, the risk assessment must be reviewed.

ISO/IEC 23894 follows a similar organizational perspective. It provides guidance for managing AI-specific risks and integrating AI risk management into the activities and functions of organizations that develop, deploy, provide, or use AI systems.

ISO/IEC 42001 further embeds AI governance in a management system that must be established, maintained, and continually improved.

The common pattern across these frameworks is clear:

AI risk management is not an approval event. It is an operating capability.

What Should Trigger a New Risk Review?

Continuous risk management does not mean that every risk must be reassessed every day. It means that reviews are built into the lifecycle and triggered by meaningful changes.

A new or updated assessment should be performed when:

- the intended use case changes,
- the solution is introduced to a new department or user group, more sensitive data is processed,
- the level of automation or autonomy increases,
- a model, provider, prompt architecture, or major component changes,
- significant performance degradation or model drift is detected,
- a security incident, complaint, near miss, or harmful output occurs,
- legal or contractual requirements change,
- the system is scaled significantly,
- or the organization considers renewing, expanding, restricting, or retiring the solution.

These triggers should be defined before the system reaches broad operational use. Otherwise, risk reviews depend on individual awareness and may happen only after an incident.

Risk Decisions Need Current Evidence

A meaningful review should not only ask whether new risks have appeared.

It should also examine:

- whether the original assumptions are still valid,
- whether the probability or impact of known risks has changed,
- whether mitigation measures were implemented,
- whether those measures are demonstrably effective,
- whether monitoring reveals unexpected behavior,
- whether users follow the intended procedures,
- and whether the remaining risk is still acceptable.

This makes **residual risk** especially important.

A mitigation measure rarely removes a risk completely. It reduces either its likelihood, its potential impact, or both. The remaining exposure must still be understood and consciously accepted by somebody with the authority to make that decision.

For example:

Human review reduces the risk of incorrect AI-generated customer responses, but it does not eliminate it. The remaining risk depends on reviewer competence, workload, process design, traceability, and the consequences of an undetected error.

The decision to continue should therefore be based on evidence, not on the existence of a control in a document.

Continuous Does Not Mean Bureaucratic

The answer to dynamic AI risks is not to create a permanent approval process for every experiment. That would replace one risk with another: the organization would become too slow to learn.

Risk management should be proportional to:

- the potential impact,
- the number of affected people,
- the sensitivity of the data,
- the autonomy of the system,
- the reversibility of possible harm,
- and the uncertainty surrounding the solution.

A low-impact internal prototype can operate with lightweight controls and rapid feedback. A solution influencing employment decisions, legal rights, safety, or critical customer processes requires stronger evidence, clearer accountability, and more formal reviews.

The objective is not maximum governance.

The objective is **the right level of governance for the current level of exposure.**

This is why AI projects need two complementary mechanisms:

- short, iterative learning cycles that reveal risks and test assumptions,
- and structured decision points that determine whether the evidence justifies increasing the organization’s exposure.

Combining Two Worlds: Agile and Stage-Gate Risk Management

AI projects need structure, but they also need room for experimentation.

A purely plan-driven approach assumes that the team can understand the problem, define the solution, predict its behavior, and identify most risks before development begins. This assumption is already problematic for complex software projects. It becomes even less realistic when the project involves uncertain data quality, probabilistic model behavior, changing providers, and users whose interaction with the system cannot be predicted completely.

However, the opposite approach is equally problematic.

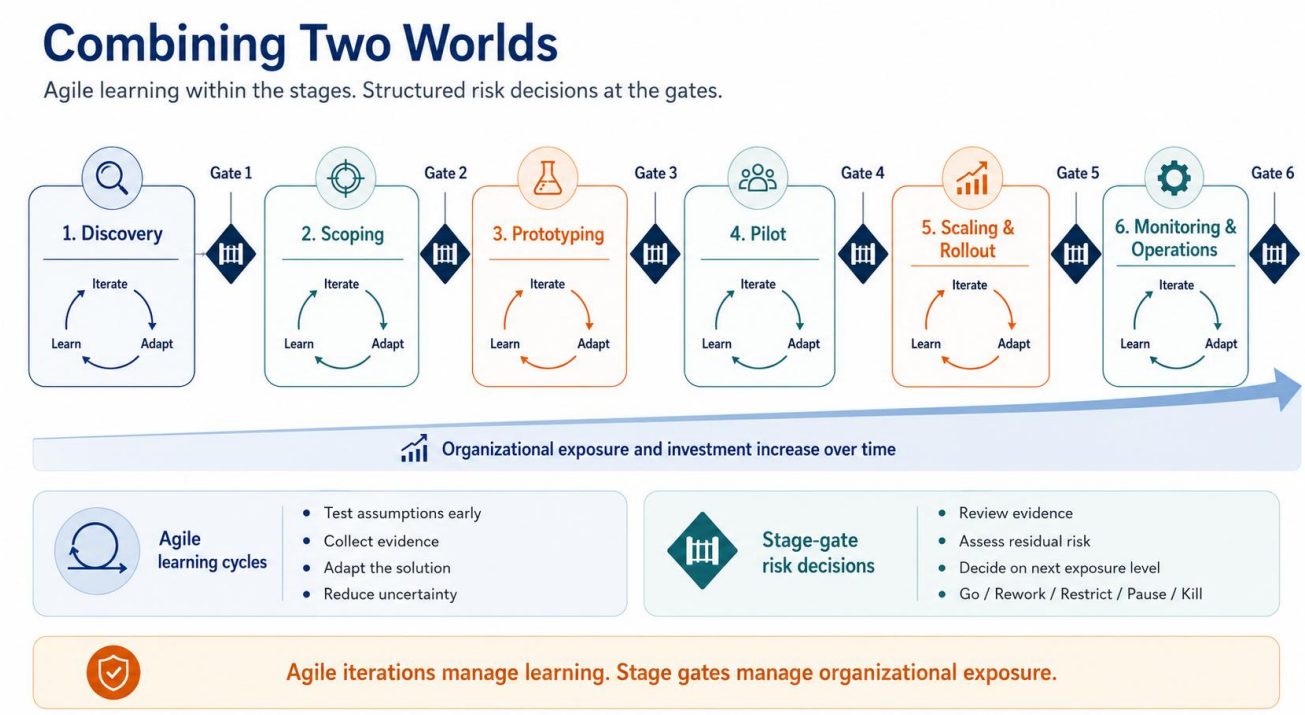
Allowing teams to experiment without clear boundaries, decision criteria, or organizational oversight can gradually increase the organization's exposure without anybody consciously deciding to accept that additional risk.

AI projects therefore need to combine two different management approaches:

- **Agile practices for learning and adaptation**
- **Stage gates for investment and risk decisions**

These approaches do not compete with each other. They solve different problems.

Agile practices reduce uncertainty through learning. Stage gates control how much exposure the organization accepts based on that learning.



Combining Two Worlds

Agile Practices Make Risks Visible

Agility is often presented primarily as a way to deliver faster.

In AI projects, its more important contribution may be that it enables teams to identify incorrect assumptions and emerging risks before they become deeply embedded in the solution.

The principles behind the Agile Manifesto emphasize early and continuous delivery, frequent working increments, collaboration with users, and adapting to changing requirements. These practices shorten the time between making an assumption and receiving evidence about whether that assumption is valid.

For AI projects, this means:

- testing a model on real or representative data early,
- involving real users before the solution is broadly deployed,
- measuring output quality instead of relying on demonstrations,
- comparing the AI approach with existing or simpler alternatives,
- testing controls and human-oversight mechanisms,
- and adapting the use case when new risks become visible.

An agile approach does not remove uncertainty. It creates a mechanism through which uncertainty can be transformed into evidence.

A team may initially assume that users will save time with an AI assistant. A small experiment may reveal that users spend more time checking the generated output than they previously spent creating it themselves.

The experiment has not failed. It has exposed a value and adoption risk before the organization invested in a full rollout.

Similarly, a prototype may reveal that the available data is insufficient, that the model cannot reliably distinguish important edge cases, or that users misunderstand the confidence of its recommendations. Each of these findings allows the team to adapt the solution while the cost and impact of change are still limited.

This is risk reduction through learning.

Stage Gates Control Increasing Exposure

Agile learning alone does not answer every management question.

A project team may be able to determine that a model performs reasonably well, users appreciate the solution, and several important risks have been reduced. But the decision to provide additional funding, connect production data, expand the user group, or allow the system to influence consequential decisions often affects more than the project team.

These decisions require broader organizational accountability.

This is the purpose of stage gates.

A gate is not simply a meeting at which the team reports what it accomplished. In the Stage-Gate model, gates are forward-looking decision points where managers assess whether a project merits additional resources and continued investment.

Applied to AI projects, each gate represents a possible increase in organizational exposure.

The next phase may involve:

- a larger budget,
- more employees and departments,
- real customer or employee data,
- connections to production systems,
- a higher level of automation,
- decisions with more serious consequences,
- or greater dependence on an external model provider.

The gate should therefore not ask only:

“Did the team complete the activities of the previous phase*?”*

It should ask:

“Does the available evidence justify the investment and risk exposure of the next phase?”

That distinction prevents gates from becoming administrative checkpoints.

A gate is a risk and investment decision, not a phase-completion ceremony.

Be Agile Within the Stages and Evidence-Driven at the Gates

The combination can be summarized in one operating principle:

Be agile within the stages and evidence-driven at the gates.

Within each phase, the team works iteratively.

It develops small increments, tests assumptions, collects feedback, measures results, and adapts the model, data, user experience, business process, or use case.

At the gate, the organization evaluates what has been learned.

The decision-makers examine:

- which assumptions were tested,
- which risks were discovered,
- which risks were reduced,
- which controls were implemented,
- whether those controls have proven effective,
- which uncertainties remain,
- and whether the residual risk is acceptable for the next phase.

This creates two connected cycles.

The Learning Cycle

The project team repeatedly:

1. builds a limited increment,
2. tests it under realistic conditions,

3. measures value, performance, and risks,
4. collects feedback,
5. and adapts the solution.

The Decision Cycle

At defined points, the responsible decision-makers:

1. review the accumulated evidence,
2. reassess the six risk domains,
3. compare the results with the gate criteria,
4. evaluate the remaining risk,
5. and decide how the initiative should continue.

This combination reflects the lifecycle-oriented logic of established AI risk frameworks. NIST's AI Risk Management Framework organizes risk management through the connected functions Govern, Map, Measure, and Manage rather than treating risk assessment as a one-time activity. The functions connect organizational governance with contextual understanding, risk measurement, prioritization, treatment, and monitoring.

For high-risk AI systems, the EU AI Act similarly defines risk management as a continuous and iterative lifecycle process requiring regular review and updating.

The proposed combination of agile cycles and stage gates translates this principle into the practical execution of an AI initiative.

What Evidence Should Be Reviewed?

A gate decision should not be based on the quality of the presentation or the confidence of the initiative owner.

It should be based on evidence.

The exact evidence depends on the phase and risk profile, but it should cover four broad perspectives.

1. Value and Business Evidence

The team should demonstrate:

- that the problem is relevant,
- that users need or benefit from the solution,
- that AI offers an advantage over realistic alternatives,
- that the expected outcome is measurable,
- and that the business case remains viable.

Early evidence may consist of interviews, process data, problem validation, or a limited experiment.

Later evidence should include observed usage, measurable outcomes, operational costs, and comparison against the previous process.

2. Data and Model Evidence

The team should demonstrate:

- that suitable data is available,
- that relevant data-quality problems are understood,
- that model performance is measured under realistic conditions,
- that important error types and limitations are known,
- and that monitoring can detect deterioration or drift.

A single performance score is rarely enough.

The evidence must show how the model behaves in the situations that matter for the intended use case.

3. Security, Compliance, Ethics, and Trust Evidence

The team should demonstrate:

- that data protection and security requirements have been addressed,
- that the intended use is legally and contractually acceptable,
- that foreseeable unfair or harmful outcomes have been examined,
- that appropriate human oversight exists,
- and that affected users understand the role and limitations of the AI system.

The required depth should be proportional to the potential impact of the system.

4. Operational and Adoption Evidence

The team should demonstrate:

- that the solution can be integrated and operated reliably,
- that ownership and support responsibilities are clear,
- that monitoring and incident procedures exist,
- that users have been trained,
- and that the system is being used as intended.

A prototype can prove that a solution is technically possible.

It cannot by itself prove that the organization is ready to operate it.

More Than Go or Kill

Gate decisions should not be limited to a binary choice between continuing and terminating the initiative.

Depending on the evidence, the decision may be:

- **Go**: The evidence justifies moving to the next phase.
- **Rework:** Important questions remain and additional learning is required.
- **Restrict:** The initiative may continue, but only within a defined use case, user group, dataset, or level of autonomy.
- **Pause:** External conditions, missing resources, or unresolved dependencies prevent a responsible continuation.
- **Kill:** The expected value no longer justifies the risk or investment.

The restrict decision is particularly useful for AI projects.

A model may be reliable enough to assist trained experts but not reliable enough to communicate directly with customers. An AI assistant may be acceptable for drafting internal content but not for producing legally binding communication. A pilot may continue with anonymized data while the use of personal data remains prohibited.

This allows the organization to preserve learning without prematurely accepting the risks of unrestricted use.

Gates Should Become Stronger as Exposure Increases

Not every phase requires the same level of governance.

During discovery, the team should be able to explore an idea with limited effort and lightweight documentation. Requiring production-level controls before the first experiment would make learning unnecessarily slow.

As the initiative progresses, however, the potential impact increases.

A useful principle is:

The strength of the gate should increase with the organization's exposure.

An early experiment may require:

- a clear problem statement,
- an initial data check,

- limited tool approval,
- and basic ethical and security boundaries.

A pilot involving real users and production data requires stronger evidence about:

- model performance,
- access controls,
- data protection,
- human oversight,
- user training,
- and incident response.

A broad rollout requires:

- operational ownership,
- scalable infrastructure,
- continuous monitoring,
- documented controls,
- support processes,
- and clear conditions for intervention or retirement.

The process therefore acts as a funnel.

Many ideas can enter the early phases at low cost. Only initiatives that generate convincing evidence and manage their risks should receive the resources and permissions required for broader use.

This Is Not Waterfall

A staged process is sometimes immediately associated with waterfall development.

But phases and gates do not determine how the team performs the work inside a phase.

A waterfall process attempts to define and complete one type of work before moving to the next. The approach described here allows continuous iteration inside every phase. Data, models, interfaces, business processes, and risk controls may be developed and tested together.

The distinction is:

- **Stages provide context and temporary objectives.**
- **Iterations generate learning inside those stages.**
- **Gates evaluate whether the accumulated evidence justifies proceeding.**

A team can run many experiments and development cycles before reaching a gate. It can also return to an earlier assumption, change the use case, narrow the scope, or repeat parts of a phase when evidence contradicts the original plan.

The process is therefore not designed to protect the plan.

It is designed to protect the organization while allowing the solution to evolve.

Structure Should Enable Innovation, Not Replace It

Poorly designed governance can become slow, political, and overly bureaucratic.

This happens when:

- gate criteria are unclear,
- the same documentation is required for every initiative,
- decision-makers are not available,
- teams prepare presentations instead of evidence,
- or committees review work without taking responsibility for decisions.

The purpose of the model is not to add as many controls as possible.

It is to create enough structure so that teams can experiment safely and decision-makers can consciously accept, reduce, or reject increasing exposure.

Well-designed guardrails create freedom.

Teams know:

- which tools and data they may use,
- which decisions require escalation,
- which evidence is expected,
- and where the non-negotiable boundaries lie.

Within those boundaries, they can move quickly.

This creates the balance that AI projects require:

Freedom to innovate within clear ethical and operational boundaries.

The next step is to turn this combination into a repeatable working cycle.

Iterate, Learn, Adapt, Decide

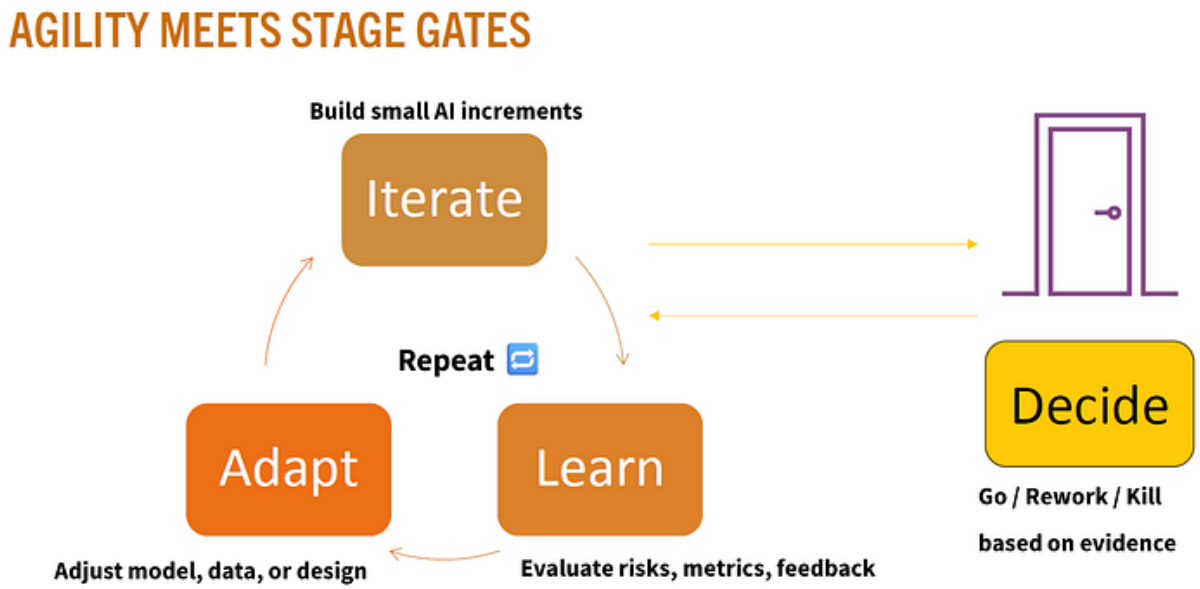
Combining agile practices with stage gates creates the overall structure for managing an AI initiative.

However, the structure only becomes useful when it changes how teams work and make decisions.

Inside each phase, the team should follow a repeatable cycle:

Iterate. Learn. Adapt. Decide. Repeat.

This cycle turns uncertainty into evidence and evidence into decisions. Instead of attempting to predict every requirement, risk, and system behavior upfront, the team builds limited increments, tests its assumptions, evaluates the results, and adjusts the solution.



Iterate. Learn. Adapt. Decide. Repeat.

The cycle is not a miniature waterfall process.

The steps do not represent separate departments or long sequential work packages. They describe four activities that may happen repeatedly within a few days, a sprint, an experiment, or an entire project phase.

1. Iterate: Build the Smallest Increment That Can Produce Useful Evidence

An iteration should not be defined only by the functionality the team wants to deliver.

It should be defined by what the team wants to learn.

Before building the next increment, the team should identify:

- the assumption it wants to test,
- the risk it wants to understand or reduce,
- the evidence required to evaluate the result,
- the boundaries within which the experiment may operate,
- and the conditions under which the experiment must be stopped.

For example, a team developing an AI assistant for customer-service employees may want to test whether the solution can generate useful draft responses.

The first iteration does not need to provide the assistant to every employee or connect it directly to customer communication. A limited experiment may use:

- a small group of trained employees,
- a restricted set of questions,

- anonymized historical cases,
- human approval before any response is used,
- and predefined quality and safety criteria.

The purpose of the iteration is not to prove that the entire solution is ready.

It is to answer a limited but important question under controlled conditions.

This approach reflects the agile principle of delivering working increments frequently and using feedback to adapt the product. In an AI context, the increment should also be designed to reveal uncertainty and test the most relevant risks.

Start with the Riskiest Assumptions

Not every assumption deserves equal attention. The team should begin with assumptions that combine:

- high uncertainty,
- high potential impact,
- and high relevance to the viability of the initiative.

These assumptions may concern different risk domains.

For example:

- **Value:** Will employees use the assistant for a real and frequent problem?
- **Business:** Does the saved time justify model, infrastructure, and review costs?
- **Data:** Is the available knowledge base complete and current enough?
- **Model:** Can the system produce reliable answers within the intended domain?
- **Operations:** Can responses, feedback, and incidents be monitored?
- **Compliance and trust:** Can sensitive information be protected, and do users understand the system's limitations?

Testing the easiest assumption first may produce quick progress, but it does not necessarily reduce the most important uncertainty.

A polished user interface is of little value when the underlying data is unsuitable or the model cannot meet the required quality threshold.

The objective of the iteration should therefore be:

Reduce the most dangerous uncertainty with the smallest responsible experiment.

Iterations Need Guardrails

Small experiments are not automatically safe experiments.

Even an early prototype may expose confidential data, produce discriminatory output, create security vulnerabilities, or influence real decisions.

The team should therefore define guardrails before the iteration begins.

Depending on the use case, these may include:

- using anonymized or synthetic data,
- restricting the experiment to a defined user group,
- prohibiting direct customer communication,
- requiring human review,
- excluding high-impact decisions,
- logging inputs and outputs,
- defining escalation procedures,
- and limiting the level of automation.

Guardrails allow the team to learn without exposing the organization to the full potential impact of an unfinished solution.

They should be strict enough to control the current risk, but not so extensive that the experiment can no longer produce meaningful evidence.

2. Learn: Evaluate Evidence, Not Demonstrations

An AI prototype can look impressive during a demonstration.

It can produce fluent text, recognize patterns, summarize documents, or generate convincing recommendations. But a successful demonstration does not prove that the solution is reliable, valuable, or ready for broader use.

Learning requires systematic evidence.

After an iteration, the team should compare:

- what it expected to happen,
- what actually happened,
- which assumptions were supported,
- which assumptions were contradicted,
- which new risks became visible,
- and which questions remain unanswered.

The evidence should cover more than technical performance.

It may include:

- observed user behavior,
- task-completion times,

- adoption and abandonment rates,
- output-quality measurements,
- false-positive and false-negative rates,
- error types and edge cases,
- human-review effort,
- infrastructure and model costs,
- security findings,
- complaints or unexpected behavior,
- and qualitative feedback from affected users.

NIST's AI Risk Management Framework distinguishes between understanding the context of an AI system, measuring its risks, and managing those risks. It also emphasizes that the four functions — Govern, Map, Measure, and Manage — are connected activities rather than a one-time checklist.

The cycle proposed in this handbook is not identical to the NIST model, but it translates the same lifecycle logic into a practical project rhythm:

- the iteration creates an opportunity to observe the system,
- learning produces evidence,
- adaptation treats the discovered risks,
- and the decision determines the next responsible step.

Negative Evidence Is Still Progress

Teams often interpret a failed experiment as a project failure.

This creates pressure to present positive results and hide contradictory evidence. In the worst case, the team gradually optimizes the presentation rather than the solution.

For risk management, negative evidence can be extremely valuable.

An experiment may show that:

- users do not trust the output,
- the model performs poorly for an important category,
- human review eliminates the expected efficiency gain,
- required data cannot be used legally,
- the solution is too dependent on one provider,
- or a simpler non-AI solution performs equally well.

These results may lead to rework, a restricted use case, or the termination of the initiative.

But they also prevent the organization from making a much larger investment based on incorrect assumptions.

The purpose of an experiment is not to confirm the initiative. It is to improve the decision.

Look Beyond Average Performance

Average metrics can hide significant risks.

A model may achieve a high overall success rate while performing poorly for:

- a specific user group,
- a particular language,
- unusual but consequential cases,
- new or changing data,
- or situations in which an incorrect result has a particularly high impact.

The team should therefore examine not only the average result, but also:

- where errors occur,
- which errors matter most,
- who is affected by them,
- whether they can be detected,
- and whether their consequences are reversible.

A model that produces an incorrect internal document summary may cause inconvenience.

A model that incorrectly rejects a customer request, recommends a personnel action, or produces unsafe technical advice may cause significantly greater harm.

Learning must therefore connect performance data with the intended context and consequences of use.

3. Adapt: Change More Than the Model

When an experiment reveals a weakness, the first reaction is often to improve the model.

The team may modify the prompt, change parameters, add more data, introduce retrieval-augmented generation, or select another model.

These may be appropriate responses, but not every AI risk is a model problem.

The team may also need to adapt:

- the intended use case,
- the user group,
- the underlying process,
- the level of automation,
- the available data,
- the user interface,
- the human-review process,
- the business model,
- the monitoring system,

- or the organizational responsibilities.

For example, a customer-service assistant may not be sufficiently reliable to send messages autonomously.

That does not necessarily mean that the entire initiative has failed.

The team could adapt the solution so that it:

- drafts responses for trained employees,
- provides source references,
- highlights uncertain statements,
- escalates specific topics,
- or operates only within a restricted knowledge domain.

The risk is not reduced by claiming that the model has improved.

It is reduced when the entire system — including technology, process, users, responsibilities, and controls — is redesigned so that the remaining risk becomes acceptable.

Adapt the Use Case When Necessary

Teams sometimes become attached to the original solution.

They continue optimizing a model even when the evidence suggests that the intended use case is unsuitable.

A responsible adaptation may therefore include:

- narrowing the scope,
- reducing the level of autonomy,
- changing from automation to augmentation,
- removing particularly sensitive decisions,
- changing the target user group,
- postponing the use of specific data,
- or replacing the AI approach with a simpler solution.

This is not a failure of agility.

It is the purpose of agility.

Responding to evidence is more valuable than protecting the original plan. The Agile Manifesto explicitly prioritizes responding to change over following a plan, and its principles call for regular reflection and adjustment of team behavior.

Controls Must Also Be Tested and Adapted

The team should not only adapt the AI solution.

It should also adapt the controls around it.

Suppose the project relies on human review to prevent incorrect customer communication. The team must examine whether:

- reviewers understand their responsibility,

- they have enough time to perform the review,
- errors are easy to recognize,
- the interface supports critical assessment,
- and the review remains effective as usage increases.

When users approve nearly every output without careful inspection, the control exists only on paper.

In this case, adaptation may require:

- better training,
- clearer source references,
- confidence indicators,
- mandatory escalation rules,
- sampling and quality reviews,
- or a more restricted use case.

A mitigation measure is itself an assumption until evidence shows that it works.

4. Decide: Convert Evidence into an Explicit Next Step

Every learning cycle should end with a decision.

Without a decision, experimentation can become endless. Teams continue improving the prototype without clarifying whether the initiative should receive more investment, remain restricted, change direction, or stop.

Decisions occur on two levels.

Decisions Within a Phase
The project team makes frequent operational decisions, such as:

- repeat the experiment with improved data,
- test another user group,
- modify the prompt or model,
- add a control,
- narrow the scope,
- or investigate a newly identified risk.

These decisions help the team determine the next iteration.

They do not necessarily require a formal gate meeting, as long as the experiment stays within the agreed scope, budget, data access, and risk boundaries.

Decisions at a Stage Gate

A stage-gate decision is required when the initiative is about to increase the organization's exposure.

Examples include:

- receiving a larger budget,
- accessing production or personal data,

- involving real customers,
- connecting to operational systems,
- expanding to additional departments,
- increasing the degree of automation,
- or moving into broad production use.

Stage-Gate guidance describes gates as forward-looking investment decisions at which an organization determines whether and how it should continue committing resources.

For an AI initiative, the decision should therefore answer:

Does the accumulated evidence justify the investment, permissions, and risk exposure of the next step?

The Five Possible Decisions

A useful decision model contains more than Go and Kill.

Go
The evidence is sufficient to proceed within the defined conditions. Known risks have been reduced to an acceptable level, required controls are in place, and ownership is clear.

Rework
The initiative remains promising, but important assumptions or controls require additional work. The team receives a clear learning objective rather than a general instruction to “improve the solution.”

Restrict
The initiative may continue, but only within explicit boundaries. Restrictions may concern:

- the permitted user group,
- available data,
- intended tasks,
- level of automation,
- geographic region,
- model provider,
- or decisions the system may influence.

Pause
The initiative cannot proceed responsibly under the current conditions. The organization may be waiting for:

- better data,
- regulatory clarification,
- infrastructure,
- contractual agreements,
- funding,
- or required expertise.

Kill
The expected value no longer justifies the remaining risk and investment. A kill decision should not be treated as an organizational failure. It may be the correct result of a successful learning process.

Every Decision Should State Its Conditions

A vague approval creates uncontrolled scope expansion. A gate should therefore not simply record:

"The pilot is approved."

It should record:

- what exactly has been approved,
- for which use case,
- for which users,
- with which data,
- under which controls,
- with which level of human oversight,
- which risks were accepted,
- who accepted them,
- and what would trigger a new review.

For example:

The assistant is approved for a three-month internal pilot with 20 trained customer-service employees. It may generate draft responses based on the approved knowledge base. All responses require human approval. Personal customer data must not be entered. Direct autonomous communication remains prohibited.

This decision is far more useful than a simple Go.

It creates a clear operating boundary for the next iteration.

Repeat: Each Decision Creates the Next Learning Cycle

The decision does not end the process.

It determines the next question.

A **Go** decision opens a new phase with greater exposure and new risks.

A **Rework** decision defines the uncertainty that must be reduced.

A **Restrict** decision creates a controlled environment in which further evidence can be collected.

A **Pause** decision identifies the conditions required before work can resume.

A **Kill** decision should capture the learning so that future initiatives do not repeat the same assumptions.

This is why the model forms a loop:

Iterate to produce evidence.
Learn from the observed results.
Adapt the complete system.

Decide whether and how to continue.
Then repeat.

A practical record for each cycle can remain simple:

Element	Question
Assumption	What do we currently believe?
Risk	What could make this assumption wrong or harmful?
Experiment	How will we test it safely?
Evidence	What did we observe?
Adaptation	What must change?
Decision	What is the next responsible step?
Owner	Who is accountable for the action or accepted risk?
Review trigger	When must the decision be reconsidered?

This structure creates traceability without turning every experiment into a large governance exercise.

It also prepares the foundation for later Decision Records, which document major choices, accepted risks, alternatives, and consequences.

The Focus of the Cycle Changes Across the Lifecycle

The cycle remains the same throughout the project:

Iterate, Learn, Adapt, Decide.

But its focus changes.
During **discovery**, it primarily tests whether the problem and opportunity justify an AI initiative.
During **prototyping**, it examines data availability, model behavior, and technical feasibility.
During the **pilot**, it evaluates real-user behavior, controls, operational integration, and measurable value.
During **scaling**, it tests whether the solution and its safeguards remain effective under broader usage.
During **operations**, it monitors whether the system continues to perform within its approved boundaries.

The next section examines how the risk profile and required evidence change across these phases.

How Risk Changes Across the AI Project Lifecycle

The six risk domains remain relevant throughout the entire AI project lifecycle. However, their importance, visibility, and potential impact change as the initiative progresses.

During **discovery**, most risks are still assumptions. The organization is primarily at risk of investing time in the wrong problem or pursuing an AI approach that creates no meaningful advantage.
During **prototyping**, data and model risks become measurable.
During a **pilot**, the system encounters real users, real processes, and real operating conditions.
During **scaling**, weaknesses that were manageable in a controlled environment may affect a much larger number of people, decisions, and business processes.

Finally, during operations, the organization must manage changes that occur after the original project team has completed its work.

The lifecycle therefore does not create six entirely different risk profiles.

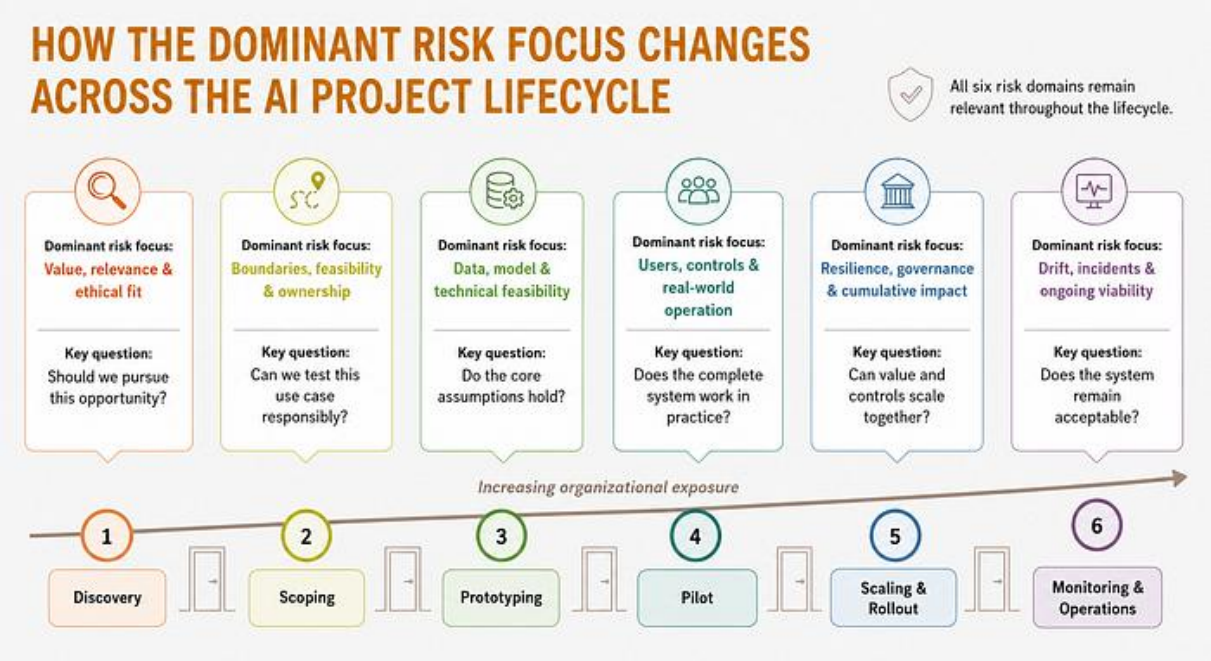
Instead, it changes:

- which risks can be evaluated,
- how reliable the available evidence is,
- how many people and systems are exposed,
- how difficult failures are to reverse,
- and how strong the required controls need to be.

NIST makes a similar distinction across the AI lifecycle. Its AI Risk Management Framework associates development with model validation and assessment, deployment with production integration and testing, and operations with ongoing monitoring, updates, incident tracking, and response. Human factors and impact-assessment activities extend across all of these stages.

The practical implication is:

Risk management must mature at the same speed as the AI solution and the organization's exposure.



1. Discovery: Is This the Right Problem for AI?

The discovery phase begins with an opportunity, a problem, or an idea.

At this point, the organization has usually invested only limited time and resources. The potential direct harm is also relatively low because no operational AI system exists yet.

But early decisions can still create significant downstream risk.

The dominant risks during discovery are:

- value and adoption risks,
- strategic alignment risks,
- initial business viability risks,
- and high-level compliance or ethical concerns.

The first question should therefore not be:

“Which model should we use?”

It should be:

“Is this a relevant problem, and does using AI create a meaningful advantage?”

Teams should examine:

- who experiences the problem,
- how important and frequent it is,
- which outcome should improve,
- how the problem is currently handled,
- whether a simpler solution could solve it,
- and what would need to be true for the AI initiative to create value.

The intended purpose and context should already be made explicit. NIST’s Map function emphasizes documenting the intended purpose, beneficial uses, applicable requirements, and the context in which the AI system may be deployed.

This is also the right time to identify obvious exclusion criteria.

Examples include:

- a use case that conflicts with company values,
- unavailable or legally unusable data,
- consequences that cannot be responsibly controlled,
- a prohibited use under applicable law,
- or an expected benefit that does not justify the likely cost.

A discovery phase should be inexpensive enough to allow exploration.

But it should still prevent the organization from investing in ideas that are fundamentally unsuitable.

Dominant risk question

Are we pursuing a relevant and responsible opportunity, or are we applying AI because the technology is available?

2. Scoping: What Exactly Are We Agreeing to Build and Test?

Once an opportunity appears promising, the initiative moves from a broad idea to a defined use case.

This transition changes the risk profile.

The organization now begins to make more specific commitments about:

- the intended users,
- the supported tasks,
- the required data,
- the model or provider,
- the business process,
- the expected outcome,
- and the boundaries of the solution.

A vague use case creates vague risks.

For example, the description:

“We want to use AI in customer service”

is too broad to assess meaningfully.

A more useful scope would define:

“The system will help trained customer-service employees draft responses to frequently asked questions based exclusively on an approved internal knowledge base. All responses require human review before they are sent.”

This scope makes several risk characteristics visible:

- The AI augments employees rather than replacing them.
- It operates within a restricted knowledge domain.
- It does not communicate autonomously.
- Human approval remains mandatory.
- The user group is known.
- The permitted data can be defined.

During scoping, all six risk domains should receive an initial assessment:

- Is the expected value measurable?
- Is the business case plausible?
- Is suitable data available?

- Are the expected model capabilities realistic?
- Can the solution be integrated and operated?
- Which legal, ethical, security, and trust requirements apply?

This is also the phase in which the organization should define:

- risk ownership,
- acceptable operating boundaries,
- initial performance expectations,
- prohibited uses,
- escalation rules,
- and criteria for the prototype.

The risk assessment remains preliminary. The team does not yet have enough evidence to prove that the system works.

But it should know which assumptions are most dangerous to be wrong about.

Dominant risk question

Is the use case defined precisely enough to test its value, feasibility, and risks responsibly?

3. Prototyping: Can the Core Assumptions Survive Contact with Reality?

The prototype phase changes the nature of the discussion.

Until this point, many claims were hypotheses:

- The necessary data is available.
- The model will be capable enough.
- The expected performance is realistic.
- The solution can be integrated.
- The risk controls will work.

A prototype creates the first opportunity to test these assumptions directly.

The dominant risk focus moves toward:

- data quality and availability,
- model and algorithm behavior,
- technical feasibility,
- provider dependencies,
- initial security controls,
- and early evidence of value.

The team should test the system under controlled but realistic conditions.

This may involve:

- representative datasets,

- known edge cases,
- adversarial or unexpected inputs,
- comparison with the current process,
- different user groups,
- and failure scenarios.

The objective is not to produce a polished demonstration.

It is to discover:

- where the model performs well,
- where it fails,
- which errors matter,
- whether errors can be detected,
- whether the available data is suitable,
- and whether the proposed controls reduce the relevant risks.

NIST recommends defining acceptable performance limits — including the distribution of errors — and establishing corrective actions when a system operates outside those limits. It also emphasizes that measurement techniques must evolve as AI systems and measurement methods change.

This is important because average performance alone can be misleading.

A system may perform well overall while failing consistently for:

- a specific language,
- a particular customer group,
- rare but high-impact cases,
- or inputs outside the original dataset.

The prototype should also reveal architectural and provider risks:

- Can the model be replaced?
- Can its output be logged?
- Can the organization control how data is processed?
- What happens when the external service is unavailable?
- Are the costs realistic under expected usage?
- Can the team reproduce and investigate problematic outputs?

At the end of prototyping, the organization should understand the system's basic capabilities and limitations much better.

It still does not know whether the system will work reliably in the organization.

That requires a pilot.

Dominant risk question

Does the prototype provide credible evidence that the core value, data, model, and technical assumptions are viable?

4. Pilot: Does the Complete System Work with Real Users?

A pilot is not merely a larger prototype. It introduces a different source of uncertainty: the real operating context. The system is now used by real people, within real processes, under realistic time pressure and organizational conditions.

This makes several risks visible that are difficult to evaluate in a laboratory:

- whether users understand the system,
- whether they use it as intended,
- whether human oversight works,
- whether the solution fits the business process,
- whether the expected value appears in practice,
- and whether operational controls remain effective.

The dominant risk focus expands toward:

- adoption and user behavior,
- calibrated trust,
- process integration,
- operational reliability,
- security and privacy,
- practical human oversight,
- and measurable business outcomes.

Consider a control requiring employees to review every AI-generated response.

On paper, this may appear sufficient.

The pilot may reveal that:

- employees do not have enough time,
- the interface makes errors difficult to detect,
- users assume that fluent responses are correct,
- responsibilities are unclear,
- or reviewers approve almost every output automatically.

The technical control exists, but the socio-technical system does not work as intended.

This is why NIST places system validation, production integration, user experience, and compliance testing in the deployment portion of the AI lifecycle. It also treats human factors as relevant throughout the lifecycle rather than as a late change-management concern.

A pilot should therefore test the complete system:

- the model,

- the data,
- the interface,
- the users,
- the process,
- the controls,
- the support structure,
- and the organizational responsibilities.

The pilot should remain deliberately restricted.

Possible boundaries include:

- a limited number of users,
- one department,
- a defined dataset,
- selected use cases,
- mandatory human approval,
- and a fixed duration.

These restrictions reduce exposure while allowing the organization to collect realistic evidence.

Dominant risk question

Does the AI solution create measurable value and remain controllable under real operating conditions?

5. Scaling and Rollout: Will the Solution and Its Controls Survive Growth?

A successful pilot does not automatically justify a broad rollout.

Scaling changes the risk profile again.

The organization may now expose:

- more users,
- more departments,
- more diverse data,
- additional regions,
- higher transaction volumes,
- more consequential decisions,
- and a larger part of its reputation and operations.

Even a small error rate can become significant when repeated at scale.

A weakness affecting one out of every thousand cases may appear negligible in a pilot involving 100 interactions. At one million interactions, the same weakness may affect approximately 1,000 cases.

Scaling therefore magnifies both value and harm.

The dominant risks now include:

- operational resilience,
- scalability and performance,
- cumulative error impact,
- cost development,
- consistent user training,
- governance across departments,
- monitoring coverage,
- support and incident response,
- and compliance across different contexts.

The organization should also examine whether the original pilot population was representative.

A system tested with a small group of highly motivated and well-trained users may behave differently when introduced to:

- less experienced employees,
- other languages,
- different processes,
- new customer groups,
- or departments with different incentives.

Controls must scale together with the solution.

For example:

- Can human review still be performed at the expected volume?
- Can incidents be detected across all departments?
- Do all users receive the necessary training?
- Is ownership still clear?
- Can support teams respond quickly?
- Can changes to prompts, data, or models be controlled?
- Are operating costs still consistent with the business case?

This phase also requires a stronger distinction between local and organizational ownership.

A pilot team may have solved problems informally. A scaled system requires documented responsibilities, repeatable processes, and durable operational capabilities.

NIST's Manage function explicitly includes determining whether an AI system achieves its intended purpose and whether its development or deployment should proceed.

The rollout decision should therefore not be based only on whether the pilot was successful.

It should ask whether the solution, controls, governance, and operating model are ready for the next level of exposure.

Dominant risk question

Can the organization scale the value of the AI solution without scaling its weaknesses beyond an acceptable level?

6. Monitoring and Operations: Does the System Remain Within Its Approved Boundaries?

The risk-management process does not end when the system goes live.

In many ways, operations reveal the most important long-term risks.

The production environment changes continuously:

- input data changes,
- user behavior changes,
- business processes evolve,
- providers update their models,
- new vulnerabilities appear,
- operating costs change,
- and the system may gradually be used beyond its original purpose.

The dominant operational risks include:

- model and data drift,
- performance degradation,
- unexpected system behavior,
- security incidents,
- provider changes,
- cost escalation,
- ineffective controls,
- use-case expansion,
- and outdated legal or organizational assumptions.

The organization must monitor both the AI system and the surrounding operating model.

This includes questions such as:

- Does the system still achieve its intended purpose?
- Are error patterns changing?
- Are certain groups or cases affected differently?
- Are users still applying the required oversight?
- Are incidents and near misses being reported?
- Are inputs and outputs changing?
- Are costs developing as expected?

- Have providers, contracts, models, or regulations changed?
- Is the system being used for tasks that were never approved?

For high-risk AI systems, the EU AI Act requires risk management to operate as a continuous, iterative process throughout the system lifecycle. It includes assessing risks arising from intended use, reasonably foreseeable misuse, and data collected through post-market monitoring.

Article 72 further requires providers of high-risk AI systems to establish a proportionate post-market monitoring system that actively and systematically collects, documents, and analyses performance data throughout the system's lifetime.

These legal requirements apply specifically to high-risk AI systems as defined by the Act. However, the underlying operating principle is valuable for AI systems more broadly:

A system that was acceptable at launch may not remain acceptable indefinitely.

Operations must therefore include clear intervention options:

- recalibrate,
- retrain,
- replace the model,
- change the process,
- strengthen controls,
- restrict usage,
- suspend the system,
- or retire it.

Continuing to operate an AI system should itself be an explicit and periodically reviewed decision.

Dominant risk question

Does the system continue to deliver value and operate safely within the conditions under which it was approved?

The Risk Focus Shifts, but No Risk Domain Disappears

The lifecycle model should not be interpreted as a handover in which one risk category ends and another begins.

Value risk does not disappear after discovery.

Data risk does not disappear after prototyping.

Compliance risk does not begin shortly before rollout.

Operational risk does not begin only after deployment.

All six domains remain relevant throughout the lifecycle.

What changes is:

- the amount of available evidence,
- the level of organizational exposure,
- the reliability of the controls,

- and the consequences of an incorrect decision.

The following summary shows the dominant focus of each phase:

Phase	Dominant risk focus	Main management question
Discovery	Value, relevance, strategic and ethical fit	Should we pursue this opportunity?
Scoping	Boundaries, feasibility, ownership and initial risk classification	Can we test this use case responsibly?
Prototyping	Data, model, technology and core assumptions	Does the solution work under controlled conditions?
Pilot	Users, adoption, controls and real-world operation	Does the complete system work in practice?
Scaling	Reach, resilience, governance and cumulative impact	Can value and controls scale together?
Operations	Drift, incidents, changes and ongoing viability	Does the system remain acceptable?

This phase-based view is a practical application of lifecycle-oriented AI risk management. It is consistent with ISO/IEC 23894's objective of integrating AI-specific risk management into organizational activities and ISO/IEC 42001's requirement to establish, maintain, and continually improve an AI management system.

It also corresponds to the six-phase project and stage-gate structure developed in this handbook.

Understanding how the risk profile changes is the first step.

The next question is more concrete:

What evidence should the team present before the organization accepts the exposure of the next phase?

Evidence Required at Each Stage Gate

A stage gate should not be a status meeting in which the project team presents completed activities.

Its purpose is to make a forward-looking decision:

Does the available evidence justify the investment, permissions, and organizational exposure of the next phase?

This distinction matters.

Stage-Gate guidance describes gates as decision points at which the organization determines whether and how it should continue investing resources. They are not intended to be retrospective project updates.

For an AI initiative, this means that completing the planned tasks is not enough.

The team may have:

- built a prototype,
- organized user interviews,
- performed a security review,
- or created a risk register.

But the gate should evaluate what these activities have demonstrated.

A completed test is an activity.

A test result showing that the model operates within an agreed error threshold is evidence.

A documented human-review process is a control.

Observed data showing that reviewers detect important errors under realistic working conditions is evidence that the control may actually work.

The gate should therefore focus on claims and evidence rather than tasks and documents.

The Evidence Must Match the Current Exposure

Not every stage gate requires production-level evidence.

During discovery, the organization cannot reasonably expect long-term operating data from a system that does not yet exist. During prototyping, it may be impossible to prove how thousands of employees will behave after rollout.

The evidence should mature together with the initiative.

Early gates primarily evaluate:

- the relevance of the problem,
- the plausibility of the opportunity,
- the initial risk profile,
- and whether further exploration is justified.

Later gates require stronger evidence about:

- model performance,

- data quality,
- user behavior,
- security,
- human oversight,
- operational resilience,
- scalability,
- and long-term viability.

This follows the same principle as proportional risk management:

The strength of the required evidence should increase with the potential impact and organizational exposure of the next phase.

NIST's AI Risk Management Framework supports tailoring risk-management activities to the organization, use case, and context. Its Playbook explicitly states that it is not a checklist that must be followed in its entirety; organizations should select and adapt the practices relevant to their situation.

A low-impact internal experiment should not require the same documentation and assurance as a system that influences employment, safety, legal rights, or customer decisions.

However, every gate should still make the next increase in exposure explicit.

What an Evidence Package Should Contain

The evidence package presented at a gate does not need to be a large report.

It should provide enough information for decision-makers to understand:

1. What the initiative currently claims
2. Which assumptions were tested
3. What evidence was collected
4. Which risks remain
5. Which controls are in place
6. Whether those controls have been tested
7. What additional exposure the next phase creates
8. Who owns the remaining risks
9. Under which conditions the initiative may continue
10. What would trigger a new review

A compact evidence package could contain:

Element	Purpose
Intended use and current scope	Clarifies what is—and is not—being assessed
Key claims and assumptions	Makes the initiative's logic explicit
Experiment and evaluation results	Shows what has actually been learned
Six-domain risk assessment	Creates a complete view of the current risks
Implemented controls	Shows how risks are being treated
Control-effectiveness evidence	Demonstrates whether controls work in practice
Residual risks	Shows which risks remain after treatment
Proposed next-phase boundaries	Defines the requested increase in exposure
Owners and accountabilities	Clarifies who accepts and manages each risk
Review and stop triggers	Defines when the decision must be reconsidered

The package should support the decision.

It should not bury the decision-makers under documentation.

Gate 1: Discovery and Opportunity Gate

The first gate determines whether an idea deserves further exploration.

At this point, the organization should not expect technical proof that the full solution works. However, it should require evidence that the opportunity is relevant enough to justify additional effort.

Required evidence

The initiative should demonstrate:

- a clearly defined user or business problem,
- evidence that the problem actually exists,
- the users or stakeholders affected,
- the expected outcome,
- alignment with organizational objectives,
- an initial explanation of why AI may be useful,
- consideration of simpler non-AI alternatives,
- preliminary data availability,
- an initial view of possible ethical, legal, and security concerns,
- and a named sponsor or accountable initiative owner.

Problem evidence may include:

- process data,

- user interviews,
- complaints,
- waiting times,
- error rates,
- manual effort,
- lost opportunities,
- or other indicators of a meaningful need.

The objective is not to prove that the proposed AI solution is correct.

It is to demonstrate that the problem is worth investigating.

NIST's AI RMF states that AI actors share responsibility for determining whether AI is an appropriate or necessary tool for a particular context.

Key gate question

Is the problem relevant enough, and is an AI approach plausible and responsible enough, to justify a structured scoping phase?

Possible decision

- **Go**: Explore and define the use case.
- **Rework**: Gather stronger problem or stakeholder evidence.
- **Restrict**: Explore only a limited or lower-risk version of the idea.
- **Kill**: The problem is not important, the AI approach offers no clear advantage, or fundamental constraints make the initiative unsuitable.

Gate 2: Scoping and Responsible-Experiment Gate

The second gate determines whether the use case has been defined precisely enough to begin technical experimentation.

A broad ambition such as "use AI to improve customer service" is not sufficient.

The organization needs to know:

- who will use the system,
- for which tasks,
- with which data,
- within which process,
- under which restrictions,
- and with what expected consequences.

Required evidence

The initiative should provide:

- a precise intended-use statement,
- explicitly excluded or prohibited uses,

- clearly defined users and affected stakeholders,
- a process and system-context description,
- an initial assessment across all six risk domains,
- preliminary data classification and usage rights,
- expected model or provider dependencies,
- measurable success criteria,
- acceptable performance and error boundaries,
- an initial business case,
- required skills, budget, and infrastructure,
- clear risk and decision ownership,
- planned prototype experiments,
- and the guardrails under which those experiments may operate.

Examples of prototype guardrails include:

- using anonymized or synthetic data,
- restricting access to a small trained team,
- prohibiting direct customer communication,
- requiring human review,
- logging all interactions,
- and excluding high-impact decisions.

NIST's Govern function emphasizes that roles, responsibilities, authority, and communication lines for AI risk management should be documented and clear.

Key gate question

Is the use case defined clearly enough, and are the initial boundaries strong enough, to test the most critical assumptions responsibly?

Possible decision

- **Go**: Begin the prototype under the approved boundaries.
- **Rework**: Clarify scope, evidence criteria, data, or ownership.
- **Restrict**: Permit only selected experiments or datasets.
- **Pause**: Wait for contracts, data access, expertise, or regulatory clarification.
- **Kill**: The use case cannot be tested responsibly or does not support a credible business case.

Gate 3: Prototype and Proof-of-Concept Gate

The prototype gate evaluates whether the initiative's core assumptions have survived technical testing.

The team should not be rewarded merely for building something that looks impressive in a demonstration.

The question is whether the prototype provides credible evidence about:

- value,
- data suitability,
- model behavior,
- technical feasibility,
- and important limitations.

Required evidence

The initiative should demonstrate:

- a working prototype,
- use of sufficiently representative test data,
- documented data-quality findings,
- measured model performance,
- relevant error distributions,
- tests of important edge cases,
- comparison with the current process or a realistic baseline,
- early feedback from intended users,
- preliminary infrastructure and usage costs,
- identified provider and architecture dependencies,
- initial security and privacy tests,
- documented model limitations,
- and evidence about whether the proposed controls work.

The model evidence should go beyond one average performance score.

It should explain:

- which types of errors occur,
- where they occur,
- who may be affected,
- how harmful they could be,
- whether they can be detected,
- and whether they are reversible.

NIST's AI RMF Core connects measurement with documented metrics, testing, evaluation, and assessment of whether an AI system performs within its intended context.

Key gate question

Does the prototype provide sufficient evidence that the initiative is technically feasible, potentially valuable, and safe enough to test with real users under controlled conditions?

Possible decision

- **Go**: Begin a restricted pilot.
- **Rework**: Improve data, model performance, architecture, or controls.
- **Restrict**: Pilot only low-impact tasks or selected user groups.
- **Pause**: Wait for better models, infrastructure, data, or vendor conditions.
- **Kill**: Core assumptions have been disproved or the remaining risk is disproportionate to the expected value.

Gate 4: Pilot and Real-World Validation Gate

The pilot gate evaluates whether the complete socio-technical system works in practice.

A model that performs well in a controlled test may behave differently when introduced into:

- real workflows,
- time pressure,
- incomplete data,
- organizational incentives,
- and human decision-making.

The pilot must therefore evaluate more than the AI component.

It must evaluate the model, data, users, process, controls, responsibilities, and operating environment together.

Required evidence

The initiative should demonstrate:

- use by a defined group of real users,
- measurable improvement over the previous process,
- observed adoption and usage behavior,
- output-quality results under real conditions,
- documented incidents and near misses,
- observed human-review behavior,
- evidence that oversight controls are effective,
- privacy and security results,
- user understanding of system limitations,
- operational logging and monitoring,
- support and escalation procedures,
- actual operating and review costs,
- updated risks across all six domains,
- and a clear assessment of residual risk.

The organization should specifically ask whether users:

- rely on the system too strongly,
- bypass required controls,
- use it outside the intended purpose,
- enter inappropriate data,
- or avoid it because it does not fit their work.

For high-risk AI systems, the EU AI Act requires effective human oversight that enables people to understand the system's capabilities and limitations, interpret its output, intervene, and stop its use when necessary.

The detailed legal obligations apply to high-risk AI systems, but the principle is valuable for AI projects more broadly: oversight should be demonstrated in practice, not merely documented.

Key gate question

Does the complete AI-enabled work system deliver measurable value and remain controllable under realistic operating conditions?

Possible decision

- **Go**: Prepare a controlled rollout.
- **Rework**: Improve workflow, training, interface, model, or controls.
- **Restrict**: Continue the pilot within narrow boundaries.
- **Pause**: Resolve operational, legal, security, or business dependencies.
- **Kill**: The system creates insufficient value or cannot be operated within acceptable risk boundaries.

Gate 5: Scaling and Rollout Gate

The rollout gate determines whether the organization is ready to increase the reach and consequences of the AI solution.

A successful pilot provides evidence about one controlled environment.

It does not automatically prove that:

- other departments will achieve the same outcome,
- controls will remain effective at higher volume,
- infrastructure will remain reliable,
- users will receive consistent training,
- or costs will remain economically viable.

Scaling changes the magnitude of both benefits and risks.

Required evidence

The initiative should demonstrate:

- repeatable value in the pilot environment,
- a rollout scope and sequence,
- validated infrastructure capacity,
- realistic cost projections at scale,
- scalable logging and monitoring,
- incident and escalation processes,
- model and provider fallback options,
- clear operational ownership,
- support and maintenance capacity,
- training and user-enablement plans,
- change-management and communication measures,
- access-control and data-governance arrangements,
- compliance documentation appropriate to the use case,
- tested business-continuity or rollback procedures,
- and defined conditions for restricting or suspending the system.

The team should also demonstrate that controls can scale.

A human-review mechanism that works for 100 outputs per week may not work for 100,000.

The same is true for:

- manual incident analysis,
- user onboarding,
- data-quality reviews,

- model evaluations,
- and approval processes.

ISO/IEC 42001 frames responsible AI governance as an organizational management system that must be established, implemented, maintained, and continually improved rather than as a one-time project activity.

Key gate question

Can the organization scale the solution, its controls, and its operating model without increasing residual risk beyond an acceptable level?

Possible decision

- **Go**: Roll out according to the approved sequence and boundaries.
- **Rework**: Strengthen operations, monitoring, training, or resilience.
- **Restrict**: Roll out only to selected teams, processes, or regions.
- **Pause**: Resolve scalability, cost, staffing, or compliance concerns.
- **Kill**: The solution cannot be operated sustainably or responsibly at scale.

Gate 6: Operational Review and Renewal Gate

The final gate is not a one-time approval at the end of the rollout.

It is a recurring decision during operations.

The organization must repeatedly determine whether the AI system:

- continues to deliver value,
- remains within its approved operating boundaries,
- and still has an acceptable risk profile.

Required evidence

Operational reviews should include:

- current performance metrics,
- changes in error patterns,
- model and data-drift indicators,
- incidents and near misses,
- user complaints and feedback,
- usage outside the approved purpose,
- human-oversight effectiveness,
- security findings,
- changes to providers or model versions,
- operating and monitoring costs,
- current adoption and business outcomes,

- changes in legal or organizational context,
- and the status of all major residual risks.

The system should also be reviewed after meaningful events such as:

- a provider model update,
- a significant incident,
- introduction to a new department,
- processing of more sensitive data,
- increased autonomy,
- or a change in the decisions the system influences.

For high-risk AI systems, Article 72 of the EU AI Act requires proportionate post-market monitoring that actively and systematically collects, documents, and analyses relevant performance data throughout the system's lifetime.

Although this obligation applies specifically to high-risk systems, the underlying principle is broadly useful:

Operational approval must be renewed through current evidence.

Key gate question

Does the AI system still deliver sufficient value and remain acceptable under its current operating conditions?

Possible decision

- **Continue**: Maintain the current approved use.
- **Improve**: Adapt model, process, data, or controls.
- **Expand**: Broaden the use only after assessing the new context.
- **Restrict**: Reduce users, tasks, data, or autonomy.
- **Suspend**: Stop operation while an issue is investigated.
- **Retire**: Decommission the system because value, viability, or trustworthiness is no longer sufficient.

The Evidence Should Become Stronger, Not Merely Larger

As the initiative moves through the lifecycle, the evidence package will usually grow.

But quantity is not the objective.

A large folder containing policies, test reports, slides, and meeting minutes can still fail to answer the essential question:

What do we now know that justifies accepting the next level of exposure?

Strong evidence is:

- relevant to the decision,
- connected to an explicit claim or risk,
- generated under realistic conditions,
- understandable to the decision-makers,
- and current enough to reflect the system being approved.

Weak evidence includes:

- demonstrations without systematic testing,
- user interest without observed usage,
- controls without proof of effectiveness,
- average performance without error analysis,
- risk registers that have not been updated,
- and approvals whose conditions are not documented.

The gate should therefore conclude with a clear decision record.

It should state:

- what was approved,
- which evidence supported the decision,
- which risks were accepted,
- who accepted them,
- which restrictions apply,
- and what would trigger a new review.

This creates the link between evidence, accountability, and traceability.

The next section examines who should provide that accountability and how risk ownership should be distributed across the organization.

Governance: Who Owns Which Risks?

AI risk management requires contributions from many different roles.

Business representatives understand the intended value and operational context. Data specialists understand the available data and its limitations. Engineers and data scientists understand the technology and model behavior. Legal, privacy, security, ethics, and employee-representation functions understand specialized constraints and potential impacts.

However, cross-functional involvement creates a common governance problem:

When everybody is involved, nobody may feel accountable.

A project team may assume that Legal owns compliance risk. Legal may assume that the business owner remains responsible for the use case. Security may approve the technical architecture without assessing whether the solution is being used appropriately. The steering committee may believe that the initiative team has already resolved all material risks.

This creates shared participation but fragmented accountability.

Good AI governance must therefore answer four different questions:

1. Who identifies and assesses the risk?
2. Who is responsible for treating and monitoring it?
3. Who has the authority to accept the residual risk?
4. Who can stop or restrict the system when conditions change?

These are not necessarily the same person.

Shared Responsibility Does Not Mean Shared Accountability

AI risks are interconnected, but each material risk still needs a named owner.

A risk owner should have:

- sufficient knowledge to understand the risk,
- the authority to coordinate mitigation measures,
- access to the required resources,
- responsibility for monitoring the risk,
- and a clear escalation path when the risk exceeds agreed boundaries.

The owner does not need to perform every control personally.

For example, a business owner may own the risk that an AI assistant provides incorrect information to customers. Technical teams may improve the model, data teams may improve the knowledge base, security teams may monitor misuse, and customer-service managers may implement human review.

But somebody must remain accountable for ensuring that these measures collectively reduce the business risk.

The phrase:

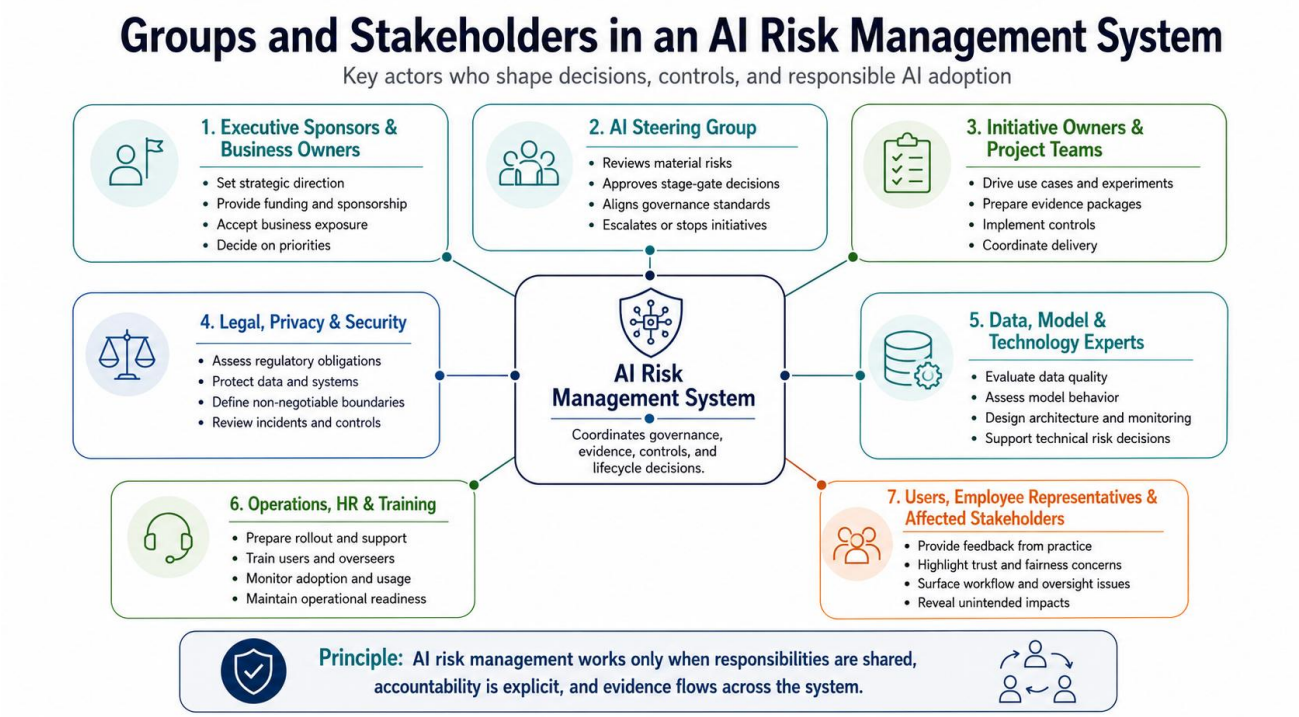
“The project team owns the risk”

is usually too vague.

Which member of the team monitors it? Who can approve additional mitigation work? Who reports a threshold breach? Who decides whether the remaining risk is acceptable?

If these questions cannot be answered, the risk does not have a real owner.

NIST's AI Risk Management Framework therefore emphasizes that organizations need explicit accountability mechanisms, clearly differentiated roles and responsibilities, appropriate incentives, and commitment from senior leadership. Its Govern function also recommends policies that distinguish the roles of people who build, use, monitor, and oversee AI systems.



Groups and Stakeholders in an AI Risk Management System

Separate Risk Ownership, Control Ownership, and Decision Authority

Three governance roles should be distinguished.

1. Risk Owner

The risk owner is accountable for managing a specific risk.

The risk owner should:

- understand the cause and potential impact,
- coordinate treatment measures,
- monitor indicators and incidents,
- keep the risk assessment current,
- and escalate when the exposure becomes unacceptable.

2. Control Owner

The control owner is responsible for operating a specific safeguard.

Examples include:

- maintaining an access-control system,

- performing data-quality checks,
- reviewing AI-generated customer responses,
- monitoring model performance,
- delivering user training,
- or operating an incident-response process.

One risk may depend on several controls with different owners.

For example, the risk of disclosing confidential information could be reduced through:

- data-classification rules,
- technical access controls,
- approved-tool policies,
- prompt filtering,
- user training,
- and monitoring.

Each control may have a different operational owner.

3. Risk-Acceptance Authority

The risk-acceptance authority decides whether the residual risk is acceptable.

This authority should match the potential impact.

A project lead may be authorized to accept a low-impact technical risk during an internal prototype. The same person should probably not be able to accept a significant privacy, employment, customer, regulatory, or reputational risk during a broad rollout.

The higher the potential impact, the more senior and cross-functional the decision authority should become.

Risk owners manage risks. Control owners operate safeguards. Authorized decision-makers accept the remaining exposure.

Without this distinction, organizations often confuse implementing a mitigation measure with formally accepting what remains.

Governance Across the Six Risk Domains

The six risk domains require different expertise and ownership. The following model is not a universal organizational chart. Titles and reporting structures differ between organizations. However, it demonstrates how ownership can be distributed without fragmenting accountability.

Risk domain	Typical primary owner	Important contributors	Typical gate authority
Value and Adoption	Business or Product Owner	Users, Change Management, UX, Department Leads	Sponsor and AI Steering Group
Business and Viability	Executive Sponsor or Business Owner	Finance, Procurement, Architecture, Vendor Management	Executive Sponsor or Investment Committee
Data	Data Owner or Data Steward	Data Engineering, Privacy, Security, Domain Experts	Data Governance and AI Steering Group
Model and Algorithm	Model Owner, Data Science Lead, or Technical Product Owner	Data Scientists, Domain Experts, Validation Specialists	Technical Authority and AI Steering Group
Technology and Operations	Service Owner or IT Operations Lead	Architecture, Security, Engineering, Support	Technology Governance and AI Steering Group
Compliance, Ethics, and Trust	Business Owner with specialist oversight	Legal, Privacy, Security, HR, Works Council, Ethics Functions	Authorized Compliance or Steering Body

The primary owner should not be interpreted as the only responsible actor.

For example, Legal can assess whether a use case complies with applicable law. But Legal cannot determine whether the business value justifies the remaining operational and reputational exposure.

Similarly, a data scientist can explain the model's measured performance and limitations. But the data scientist should not independently decide whether that performance is acceptable for a consequential business process.

The final judgment requires business context.

A 5% error rate has a different meaning when the system:

- summarizes internal meeting notes,
- drafts marketing content,
- recommends customer actions,
- supports a medical decision,
- or evaluates job applicants.

Technical experts provide evidence.

Business owners and authorized governance bodies decide whether the evidence is sufficient for the intended context.

The Initiative Owner Manages the Complete Risk Picture

Every AI initiative should have one clearly named initiative owner.

This role may be called:

- AI Initiative Owner,
- Product Owner,
- Business Owner,
- Use-Case Owner,
- or AI Product Manager.

The title matters less than the mandate.

The initiative owner should maintain the integrated view across all six risk domains.

This includes:

- keeping the intended use and scope clear,
- maintaining the current risk assessment,
- coordinating the domain experts,
- ensuring that mitigation actions are completed,
- collecting evidence for stage-gate decisions,
- documenting unresolved risks,
- and escalating when the project exceeds approved boundaries.

The initiative owner does not replace Legal, Security, Data Governance, or technical experts.

The role connects their findings.

Without this integration, the initiative may pass several specialized reviews while still presenting an unacceptable overall risk.

For example:

- Security approves the architecture.
- Legal confirms that the proposed data use is permissible.
- Data Science demonstrates acceptable model performance.
- Finance confirms that the pilot is affordable.

But the initiative may still fail because users do not understand the system, the human-review process is ineffective, or the expected benefit disappears once review costs are considered.

Someone must combine the evidence into a system-level judgment.

That is the responsibility of the initiative owner.

Domain Experts Assess Risks but Should Not Become Approval Bottlenecks

Specialist functions play an essential role in AI governance.

Depending on the use case, these may include:

- Legal,
- Data Protection,
- Information Security,
- Data Governance,
- Enterprise Architecture,
- Finance and Controlling,
- Procurement and Vendor Management,
- HR,
- Works Council or employee representatives,
- Ethics or ESG functions,
- and domain specialists.

Their purpose is not merely to approve or reject an initiative.

They should help the team:

- identify risks early,
- define suitable guardrails,
- design meaningful experiments,
- establish evidence requirements,
- and determine when escalation is necessary.

This distinction matters.

When experts are consulted only shortly before rollout, they often have only two options:

- accept a solution whose important design decisions have already been made,
- or stop a project after substantial investment.

Early participation allows them to shape the experiment instead.

A privacy specialist may recommend using anonymized data during prototyping. Security may permit a limited test using an approved platform. Employee representatives may help define acceptable workplace use and transparency. Legal may identify a narrower use case with a lower regulatory burden.

Good governance turns experts into design partners rather than late-stage gatekeepers.

However, specialist involvement must remain proportional.

Not every low-impact prototype requires a meeting with every corporate function. The level of involvement should depend on:

- the sensitivity of the data,
- the potential impact,
- the intended users,
- the autonomy of the system,
- the reversibility of possible harm,

- and the uncertainty surrounding the solution.

NIST recommends allocating oversight and risk-management resources in relation to the organization's risk tolerance: lower tolerances and higher-risk systems warrant stronger mitigation and oversight.

The AI Steering Group Controls Exposure, Not Daily Delivery

The AI Steering Group has a different responsibility from the initiative team.

It should not manage the daily development of the solution.

Its role is to:

- define organizational AI principles and risk tolerance,
- establish common gate criteria,
- decide which initiatives receive additional resources,
- resolve cross-functional conflicts,
- review material residual risks,
- approve increases in organizational exposure,
- and restrict or stop initiatives when necessary.

A useful steering group combines three complementary perspectives:

Economic and Business Alignment

Typical participants:

- executive sponsors,
- business leaders,
- finance or controlling,
- and portfolio representatives.

They ask:

- Does the initiative support strategic objectives?
- Is the expected value credible?
- Is the required investment justified?
- Is the business case sustainable?
- Does the initiative deserve priority over alternatives?

Legal, Privacy, and Security

Typical participants:

- Legal,
- Data Protection,
- Information Security,

- Risk Management,
- and relevant technology governance roles.

They ask:

- Is the proposed use permissible?
- Are data and systems sufficiently protected?
- Which obligations and controls apply?
- Are documentation and monitoring sufficient?
- Which risks cannot be accepted?

Ethics, People, and Trust

Typical participants:

- HR,
- Works Council or employee representatives,
- ethics or ESG representatives,
- Change Management,
- and representatives of affected user groups.

They ask:

- Who could be affected?
- Is the system fair and transparent?
- Is human oversight meaningful?
- Are employees adequately informed and trained?
- Could the solution undermine trust or create inappropriate pressure?

The steering group does not need to contain every specialist permanently.

A smaller standing group can invite specialists based on the risk profile of the initiative.

The critical point is that the group has:

- a defined mandate,
- named decision-makers,
- clear escalation thresholds,
- and the authority to approve, restrict, pause, or terminate initiatives.

A committee that can only make recommendations does not truly control risk exposure.

Executive Sponsors Cannot Delegate Accountability Completely

Every significant AI initiative needs an executive sponsor.

The sponsor ensures that the initiative:

- has a legitimate strategic purpose,

- receives appropriate resources,
- has access to relevant stakeholders,
- and is not abandoned between organizational boundaries.

The sponsor also carries responsibility for the economic and organizational logic of the initiative.

An executive sponsor should not approve a project simply because technical and compliance specialists have raised no objections.

The sponsor should understand:

- what value the initiative is expected to create,
- which major risks remain,
- which restrictions apply,
- who is affected,
- what the next phase will expose,
- and under which conditions the initiative will be stopped.

Specialists can assess parts of the risk profile.

They cannot replace executive accountability for the overall decision.

ISO/IEC 42001 frames AI governance as an organizational management system rather than a technical project process. This includes defined responsibilities, accountability, risk assessment, transparency, and continual improvement.

Human Oversight Requires Competence and Authority

Many AI governance models include the phrase:

“A human remains in the loop.”

But the presence of a person does not automatically create effective oversight.

The person must be able to:

- understand what the system can and cannot do,
- recognize when output may be unreliable,
- challenge the recommendation,
- override or reject it,
- escalate concerns,
- and stop the process when necessary.

Human oversight fails when the assigned person:

- lacks sufficient training,
- has no time to review the output,
- cannot access relevant context,
- is pressured to follow the recommendation,
- or lacks the authority to intervene.

For high-risk AI systems, the EU AI Act requires deployers to assign human oversight to people with the necessary competence, training, authority, and support. Article 14 also requires systems to enable people to monitor, interpret, override, or stop their operation and to remain aware of potential over-reliance on AI output.

Although these specific legal requirements apply to high-risk systems, they express a broadly useful governance principle:

Responsibility without competence and authority is not meaningful oversight.

The human overseer should therefore be treated as a defined operational role with:

- training requirements,
- decision rights,
- workload assumptions,
- escalation paths,
- and measurable control effectiveness.

Legal Roles and Internal Roles Must Be Mapped Explicitly

Organizations also need to distinguish between internal project roles and the legal roles defined in regulation.

Under the EU AI Act, a company may act as a:

- provider,
- deployer,
- importer,
- distributor,
- product manufacturer,
- or another party in the AI value chain.

These legal roles create different obligations.

For high-risk AI systems, providers are responsible for matters such as compliance with system requirements, quality management, technical documentation, logging, conformity assessment, and corrective action. Deployers must use systems according to instructions, assign competent human oversight, monitor operation, manage relevant input data, retain logs where required, and report certain risks or incidents.

The internal label “project owner” does not answer which legal role the organization occupies.

The mapping becomes particularly important when an organization:

- substantially modifies an acquired AI system,
- changes its intended purpose,
- integrates it into a larger product,
- or markets it under its own name.

Under certain conditions, a deployer or another third party can become the provider and assume the corresponding obligations.

The initiative should therefore document:

- which external providers are involved,
- which legal role the organization has,
- whether that role could change,
- which obligations follow from the role,
- and which internal owner is responsible for satisfying them.

This assessment should be performed by qualified legal and compliance specialists. It should not be inferred solely from project terminology.

Users and Affected Stakeholders Are Part of Governance

Governance should not consist only of managers and specialists reviewing the initiative from outside.

Users and affected stakeholders provide evidence that cannot be generated by the project team alone.

They can reveal:

- whether the solution fits the real process,
- whether explanations are understandable,
- whether controls work under time pressure,
- whether the system changes behavior or incentives,
- whether people feel unfairly treated,
- and whether unanticipated harms or workarounds emerge.

Their participation may take different forms:

- user research,
- pilot feedback,
- employee-representation involvement,
- impact assessments,
- complaints and appeal mechanisms,
- incident reporting,
- and periodic operational reviews.

This does not mean that every stakeholder receives a veto over every technical decision.

It means that decisions affecting people should not be based exclusively on assumptions made about them.

NIST treats AI risk as socio-technical and recommends communication between AI actors, executive leadership, users, and impacted communities, as well as audits of whether these accountability mechanisms are effective.

A Practical Accountability Model

For each material AI risk, the initiative should document at least:

Governance field	Question
Risk owner	Who is accountable for managing this risk?
Control owner	Who operates each mitigation measure?
Evidence owner	Who collects and validates the relevant evidence?
Acceptance authority	Who may accept the residual risk?
Escalation path	Who must be informed when thresholds are exceeded?
Stop authority	Who can restrict or suspend the system?
Review trigger	Which change or event requires reassessment?
Affected stakeholders	Who needs to be consulted, informed, or protected?

Example:

Field	Example
Risk	Incorrect AI-generated advice is sent to customers
Risk owner	Head of Customer Service
Control owners	AI Product Team and Customer Service Quality Lead
Evidence owner	Quality Assurance Lead
Acceptance authority	Customer Operations Director and AI Steering Group
Escalation trigger	Critical error or error rate above agreed threshold
Stop authority	Service Owner or Customer Operations Director
Affected stakeholders	Customer-service employees and customers

This creates a much stronger accountability model than simply assigning the risk to “IT” or “the project.”

Common Governance Anti-Patterns

The Steering Group Owns Every Risk

When the steering group formally owns all AI risks, initiative teams may stop feeling accountable for daily management.

The steering group should control exposure and accept material residual risks. It should not become the operational owner of every mitigation action.

Legal Approval Is Treated as General Approval

Legal approval answers specific legal questions.

It does not prove:

- business value,
- model reliability,
- operational readiness,
- adoption,
- or overall trustworthiness.

Compliance is necessary, but it is not sufficient.

Accountability Without Authority

A person is named as risk owner but has no budget, decision rights, or ability to change the system.

This creates symbolic accountability.

Human Oversight Exists Only on Paper

A person is expected to review output but lacks the competence, time, information, or authority to intervene.

Committees Review but Nobody Decides

The initiative is discussed repeatedly, but no participant has the authority to approve, restrict, or stop it.

Specialists Are Involved Too Late

Legal, security, privacy, ethics, or employee representatives encounter the solution only after the main design decisions have been made.

Risks Are Distributed by Department Instead of Managed as a System

Each function reviews its own area, but nobody examines how data, model, process, users, controls, and business consequences interact.

Governance Turns Evidence into Accountability

AI risk governance should not be designed to move every decision upward.

Its purpose is to place each decision at the appropriate level.

Teams should be able to experiment rapidly within approved boundaries.

Domain experts should help identify risks and design controls.

Risk owners should monitor the current exposure.

The steering group should decide when the organization is prepared to accept more investment and broader consequences.

Executive sponsors should remain accountable for the overall business decision.

This creates a distributed but explicit governance system:

Decisions stay close to the work until the initiative asks the organization to accept more exposure.

However, a governance decision is only useful when the organization can later understand:

- what was decided,
- why it was decided,
- which alternatives were considered,
- which risks were accepted,
- and under which conditions the decision remains valid.

This requires traceability.

Decision Records, Tech Radar, and Traceability

A good stage-gate decision answers an immediate question:

Should the AI initiative move forward under the proposed conditions?

However, this decision may need to be understood months or years later.

The original project team may no longer exist. The model may have changed. New users may apply the system in a different context. A provider may release another version or change its contractual conditions. An incident may raise questions that nobody anticipated during the original review.

At that point, the organization must still be able to understand:

- what was decided,
- why the decision was made,
- which alternatives were considered,
- which evidence was available,
- which risks were accepted,
- which restrictions applied,
- and under which conditions the decision should be reviewed.

Without this traceability, governance gradually becomes organizational memory.

People remember that a model, tool, or use case was once "approved," but they no longer remember what exactly was approved or which assumptions supported the decision.

This creates a dangerous simplification:

"The AI Steering Group approved it."

But approval is rarely unconditional.

A pilot may have been approved only for:

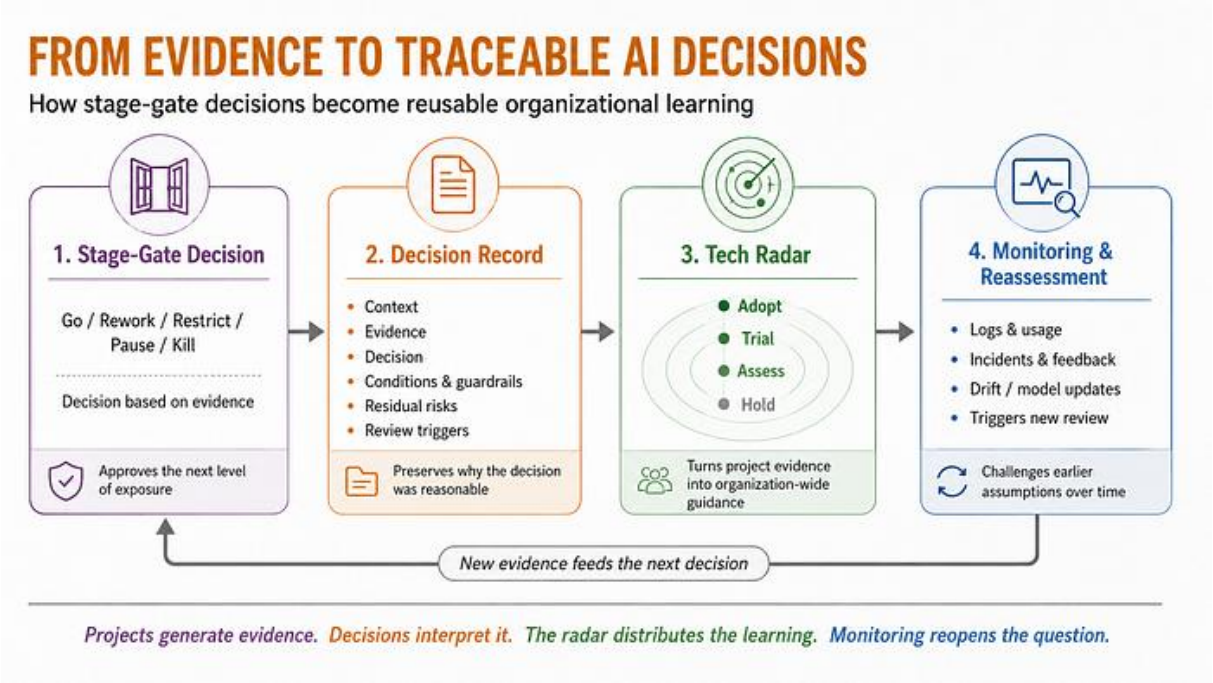
- one department,
- a limited group of trained users,
- anonymized data,
- a specific model version,
- mandatory human review,
- and a defined period.

If these conditions disappear from organizational memory, a limited approval can gradually be interpreted as unrestricted permission.

Risk governance therefore needs more than meetings and verbal agreements.

It needs traceable decisions.

Traceability is not created by a single document or governance tool. It emerges from a connected system in which stage-gate decisions, Decision Records, the Tech Radar, and operational monitoring reinforce one another.



Stage-gate decisions determine the next level of exposure. Decision Records preserve the reasoning, the Tech Radar distributes the learning across the organization, and monitoring creates new evidence for reassessment.

Decision Records Preserve the Reasoning

Architecture Decision Records are commonly used in software development to document significant architectural choices. AWS describes an ADR as a record containing the decision, its context, and its consequences. ADRs also have a lifecycle and can later be superseded when circumstances or decisions change.

The same principle can be extended beyond architecture.

For AI initiatives, the organization should document major decisions relating to:

- intended use,
- model selection,
- provider selection,
- data usage,
- level of automation,
- human oversight,
- performance boundaries,
- risk acceptance,
- pilot restrictions,
- rollout approval,
- and system retirement.

These records could be called:

- AI Decision Records,
- Risk Decision Records,
- Governance Decision Records,
- or simply Decision Records.

The name is less important than the purpose:

A Decision Record explains why a significant AI decision was reasonable under the conditions and evidence available at that time.

A Decision Record should not attempt to document every conversation or minor implementation choice.

It should capture decisions that materially affect:

- the system's risk profile,
- the organization's exposure,
- future technology options,
- legal responsibilities,
- operational ownership,
- or the ability to reverse the decision.

What an AI Decision Record Should Contain

A practical AI Decision Record can remain relatively short.

It should contain enough information to reconstruct the decision without becoming a large approval document.

Fields and Purpose

Field	Purpose
Title	Identifies the decision clearly
Status	Proposed, accepted, rejected, superseded, or retired
Date and decision authority	Shows when and by whom the decision was made
Context	Explains the problem and current situation
Intended use	Defines what the AI system is supposed to do
Excluded uses	Defines what remains prohibited
Options considered	Shows which realistic alternatives were evaluated
Evidence	Summarizes the tests, metrics, and findings used
Risk assessment	Documents relevant risks across the six domains
Decision	States what was approved, rejected, or restricted

Field	Purpose
Conditions and guardrails	Defines the boundaries of the decision
Residual risks	Records which risks remain after mitigation
Risk acceptance	States who accepted the remaining exposure
Consequences	Describes expected benefits, costs, dependencies, and trade-offs
Monitoring indicators	Defines what must be observed during operation
Review triggers	Defines when the decision must be reconsidered
Related records	Links to risk registers, tests, models, contracts, or earlier decisions

A Decision Record should make the scope of approval explicit.

Instead of writing:

"The use of Model X was approved."

write:

"Model X was approved for a three-month internal pilot that generates draft responses for trained customer-service employees. Customer communication requires human approval. Confidential customer data may not be entered. The decision must be reassessed after a significant model update, provider-contract change, serious incident, or expansion of the use case."

The second version is more useful because it makes the boundaries and review conditions visible.

Record the Decision, Not Only the Successful Option

A strong Decision Record should also capture the alternatives that were rejected.

For example, a team may compare:

- developing an internal model,
- using a managed external model,
- using a retrieval-augmented architecture,
- relying on a deterministic search function,
- or improving the existing process without AI.

Recording the alternatives helps future teams understand why the chosen option appeared reasonable.

Without this context, a later team may reopen the same discussion without knowing that the alternatives were already evaluated.

AWS notes that ADRs can support team alignment, communicate significant choices, and prevent teams from repeatedly discussing the same architectural questions.

However, previously rejected alternatives should not be treated as permanently invalid.

A different model, lower costs, better data, new regulation, or another operating context may change the decision.

This is why Decision Records need a status and lifecycle.

Do Not Rewrite History

When circumstances change, organizations are sometimes tempted to update the old record so that it reflects the new decision.

This destroys traceability.

The original record should remain available because it shows what the team knew and why the original decision was made.

A new Decision Record should:

- reference the previous one,
- explain what changed,
- describe the new evidence,
- and mark the old decision as superseded or retired.

AWS recommends preserving ADR history rather than deleting earlier decisions, including keeping superseded records and documenting ownership of changes.

This creates an important distinction:

- An incorrect decision is a decision that was unreasonable based on the available evidence.
- An outdated decision may have been reasonable originally but no longer fits the current conditions.

AI governance needs to recognize this difference.

The purpose of traceability is not to assign blame to earlier decision-makers.

It is to make learning visible.

The Tech Radar Makes Organizational Risk Appetite Visible

Decision Records describe individual choices.

A Tech Radar serves a different purpose.

It creates an organization-wide view of technologies, tools, platforms, models, and practices and communicates how the organization currently evaluates them.

The Thoughtworks Technology Radar uses four rings:

- **Adopt**
- **Trial**
- **Assess**
- **Hold**

The rings communicate increasing levels of confidence or caution. “Adopt” indicates technologies that should be seriously considered. “Trial” includes technologies considered ready for limited use but not yet as proven. “Assess” contains technologies worth examining, while “Hold” indicates technologies that should generally be approached with caution or avoided for new work.

An internal AI or Technology Radar can use the same logic to make the organization’s current position transparent.

It might include:

- foundation models,
- AI platforms,
- coding assistants,
- retrieval frameworks,
- vector databases,
- evaluation tools,
- observability platforms,
- agent frameworks,
- approved AI development patterns,
- and high-risk practices the organization wants to avoid.

The Radar should not simply indicate which technologies are popular.

It should communicate the organization's current combination of:

- experience,
- strategic relevance,
- operational readiness,
- security confidence,
- compliance status,
- and risk appetite.

What the Four Rings Could Mean for AI

Adopt
The technology or practice has sufficient organizational evidence for approved use.

This does not necessarily mean unrestricted use.

The Radar entry should still define:

- approved use cases,
- data restrictions,
- available support,
- required controls,
- and relevant Decision Records.

Trial
The technology may be used in controlled initiatives.

Usage should be limited, monitored, and connected to explicit learning objectives.

This ring is especially useful for AI technologies because it creates room for experimentation without treating a successful prototype as a general endorsement.

Assess
The technology is relevant enough to investigate, but the organization does not yet have sufficient evidence to use it in real operational contexts.

Activities may include:

- technical evaluation,
- security assessment,
- legal review,
- cost analysis,
- or controlled experiments with synthetic data.

Hold
The organization should not begin new initiatives with this technology or practice under current conditions.

Possible reasons include:

- unresolved security concerns,
- unclear licensing,
- unacceptable provider dependency,
- poor model performance,
- insufficient transparency,
- regulatory uncertainty,
- unavailable support,
- or an alternative technology that better fits the organization’s strategy.

“Hold” does not always mean that a technology is inherently bad.

It may mean:

The technology currently exceeds the organization’s risk tolerance or does not fit its strategy.

A Radar Entry Needs More Than a Name

A useful Radar entry should contain more than the name of a model or tool.

It should include:

- technology or model name,
- current version where relevant,
- ring,
- date of assessment,
- responsible owner,
- approved and excluded use cases,
- data classification restrictions,
- known risks and limitations,
- required controls,
- links to relevant Decision Records,
- and the next planned review.

For example:

Field	Example
Technology	External generative AI model
Ring	Trial
Approved use	Internal document summarization using non-confidential data
Prohibited use	Autonomous customer communication and personnel decisions
Required controls	Human review, logging, approved interface
Main risks	Hallucination, provider dependency, data exposure
Decision Record	ADR-AI-014
Review trigger	Major model update or new contractual terms
Owner	AI Platform Lead

This prevents the Radar from becoming a decorative collection of logos.

Decision Records and the Tech Radar Solve Different Problems

The Tech Radar and Decision Records should complement each other.

Instrument	Primary purpose
Stage Gate	Decides whether an initiative may increase its exposure
Risk Register	Maintains the current risks, controls, owners, and status
Decision Record	Documents why a significant decision was made
Tech Radar	Communicates the organization's strategic position toward technologies and practices
Monitoring and Logs	Provide operational evidence about actual system behavior
Policies and Guardrails	Define the boundaries within which teams may act

The connections matter.

A stage-gate decision may approve an initiative using a technology currently classified as **Trial**.

The Decision Record documents why that limited exception is reasonable.

The Tech Radar communicates that the technology remains restricted for the wider organization.

Monitoring then provides the evidence needed to decide whether the technology should later move to:

- Adopt,
- remain in Trial,
- return to Assess,
- or move to Hold.

This creates an organizational learning loop:

Projects generate evidence. Decisions interpret the evidence. The Radar distributes the learning across the organization. Monitoring challenges the decision over time.

Traceability Must Connect Decisions to the Actual System

A Decision Record is valuable only when it can be connected to the system that was actually built and operated.

The record should link to relevant artifacts such as:

- model and provider versions,
- source-code repositories,
- prompt configurations,
- datasets and data lineage,
- architecture diagrams,
- evaluation reports,

- security assessments,
- risk registers,
- contracts,
- operating procedures,
- monitoring dashboards,
- and incident records.

NIST's AI RMF Playbook recommends documenting processes for risk mapping and measurement, connecting AI governance with broader organizational controls, tracking third-party and pre-trained model risks, and maintaining sufficient records for governance and monitoring.

Traceability should enable the organization to answer:

- Which system version was evaluated?
- Which model produced the output?
- Which data was used?
- Which controls were active?
- Which decision allowed this use?
- Which person or role approved the remaining risk?
- Which later changes may have invalidated the decision?

A document that cannot be linked to the deployed system creates administrative traceability, but not operational traceability.

Technical Logs Are Not the Same as Decision Records

Technical logs record what happened during system operation.

Decision Records explain why the organization chose to operate the system in a particular way.

Both are needed.

Logs may show:

- when a model was called,
- which system version was active,
- which user initiated the action,
- whether a human reviewed the result,
- and which error or incident occurred.

The Decision Record explains:

- why the system was allowed to perform that action,
- which risks were known,
- which restrictions applied,
- and who accepted the residual exposure.

For high-risk AI systems, Article 11 of the EU AI Act requires technical documentation to be created before the system is placed on the market or put into service and kept up to date. Article 12 requires technical capabilities for automatic event logging throughout the system lifecycle to support traceability, risk identification, operational monitoring, and post-market monitoring. These legal requirements apply specifically to high-risk AI systems, but they illustrate the broader distinction between documenting system design and recording actual operation.

Providers of high-risk AI systems also have specific documentation-retention and log-retention obligations under Articles 18 and 19.

Internal Decision Records do not automatically satisfy these legal requirements.

They are an organizational governance instrument. Where the AI Act or other laws apply, qualified legal and compliance experts must determine the required technical documentation, logging, retention, and conformity evidence.

An Example: Selecting a Model for a Customer-Service Assistant

Consider an initiative developing an AI assistant for customer-service employees.

The team evaluates two external models and an internal open-source alternative.

Context
The assistant should draft responses using an approved knowledge base. Employees remain responsible for reviewing and sending the final message.

Options

- Provider A: strongest measured quality, but higher cost and provider dependency.
- Provider B: slightly lower quality, but stronger contractual controls and regional hosting.
- Internal model: greater control, but substantially higher development and operational effort.

Evidence
The team tests:

- answer quality,
- hallucination rates,
- response time,
- data-handling conditions,
- infrastructure requirements,
- provider contracts,
- and operating costs.

Decision
Provider B is approved for a limited pilot.

Conditions

- No autonomous customer communication.
- Mandatory human review.
- Only the approved knowledge base may be used.
- Sensitive customer data must be masked.

- Inputs and outputs must be logged according to policy.
- The model may not be used for legal or financial commitments.

Residual risks

- Provider dependency remains.
- Incorrect answers may not always be detected by reviewers.
- Costs may rise with higher usage.

Review triggers

- Significant model update.
- Contractual or pricing change.
- Critical customer incident.
- Expansion to another language or department.
- Proposal to remove human review.

The Decision Record preserves this reasoning.

The Tech Radar may then list Provider B as Trial, restricted to approved internal assistance use cases.

The pilot generates new evidence. Based on the results, the Radar position and the initiative decision can later change.

Common Traceability Anti-Patterns

Approval Without Conditions
A tool or model is labelled "approved," but the organization does not document the approved use case, data restrictions, or required controls.

The Radar Becomes a Logo Collection
Technologies are placed in rings without clear reasoning, owners, boundaries, or review dates.

Decision Records Are Written After the Decision
Documentation is created retrospectively to justify an outcome rather than to capture the real reasoning and alternatives.

Every Minor Choice Gets a Decision Record
Teams document so many small decisions that important strategic and risk-related choices become difficult to find.

Old Decisions Are Deleted
Superseded records disappear, making it impossible to reconstruct why the system evolved.

Documentation Is Disconnected from Operations
The record refers to a model, dataset, or control, but the organization cannot determine whether the deployed system still uses them.

"Adopt" Is Interpreted as Universal Permission
A technology's Radar classification is treated as approval for every use case, regardless of data, context, users, or potential impact.

Monitoring Does Not Feed Back into Governance
Incidents, drift, model changes, and user behavior are observed, but they do not trigger a new Decision Record, risk review, or Radar reassessment.

Traceability Creates Organizational Learning

The purpose of Decision Records and a Tech Radar is not to produce more documentation.

It is to preserve and distribute learning.

Decision Records ensure that significant choices remain understandable.

The Tech Radar turns project-specific evidence into shared organizational guidance.

Risk registers maintain the current exposure.

Logs and monitoring challenge earlier assumptions with real operational evidence.

Together, these mechanisms allow the organization to move faster without repeatedly starting from zero.

Teams can see:

- which tools have already been evaluated,
- which risks were discovered,
- which controls worked,
- which decisions remain valid,
- and where caution is required.

Traceability therefore supports both governance and innovation.

Transparent decisions create reusable knowledge. Reusable knowledge allows future teams to move faster with greater confidence.

However, teams should not need to consult a committee or read several records before every experiment.

They also need clear, practical boundaries for everyday work.

Practical Guardrails for AI Projects

Governance defines who may make decisions.
Stage gates determine whether an initiative may increase the organization's exposure.
Decision Records preserve why those decisions were made.

But employees and project teams still need to make hundreds of smaller decisions between the gates.

They need to know:

- which AI tools they may use,
- which data they may process,
- which activities require human review,
- when they need specialist approval,
- what must be documented,
- and what to do when the system behaves unexpectedly.

A steering group cannot decide each of these questions individually.

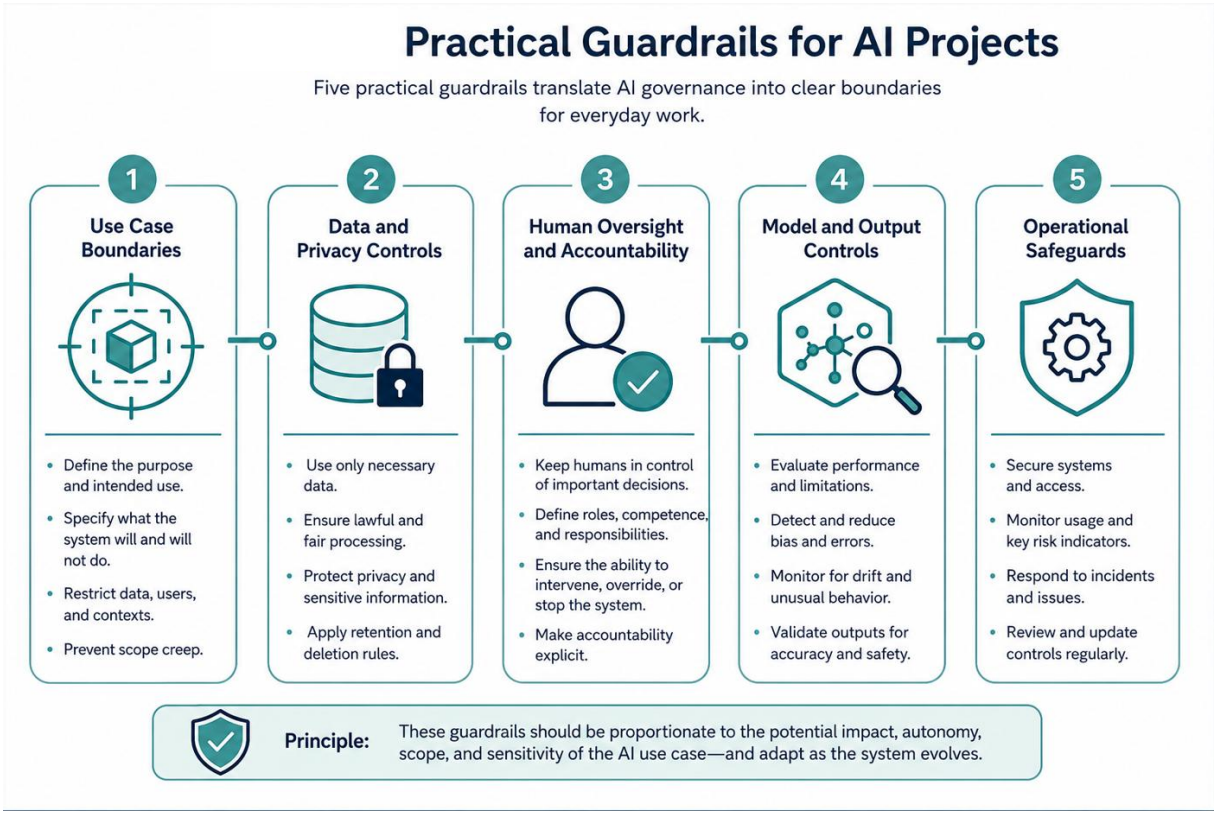
This is the purpose of guardrails.

Guardrails translate organizational principles and risk decisions into practical boundaries for everyday work.

Good guardrails do not prescribe every development step. They define the limits within which teams may experiment and make decisions independently.

They should make three things clear:

1. What teams may do without additional approval
2. What teams may do only under defined conditions
3. What teams must not do or must escalate



Five practical guardrails turn AI governance into clear operating boundaries for everyday work. Their strength should increase with the impact, autonomy, scope, and data sensitivity of the use case.

Guardrails therefore differ from detailed procedures.

A procedure explains exactly how an activity should be performed. A guardrail defines the boundary that the activity must not cross.

For example:

- A procedure may explain how to anonymize a dataset.
- A guardrail may state that personal customer data must not be entered into an external AI service unless the service and use case have been explicitly approved.

This distinction preserves flexibility.

Teams can choose their implementation approach as long as they remain within the approved operating boundaries.

Guardrails Should Be Risk-Based

Not every AI use requires the same restrictions.

An employee using an approved assistant to improve the wording of a non-confidential internal document presents a different risk from a system that:

- processes customer data,
- recommends personnel decisions,
- communicates autonomously,
- or influences legal rights.

Guardrails should therefore become stronger as the potential impact increases.

A practical model may distinguish between three operating levels.

Level	Typical use	Example guardrails
Low exposure	Individual productivity using non-sensitive data	Approved tool, no confidential data, user remains accountable
Controlled exposure	Team or departmental use involving business processes	Registered use case, logging, defined ownership, human review
High exposure	Consequential decisions, sensitive data, broad automation	Formal risk assessment, specialist reviews, stage-gate approval, continuous monitoring

The exact classification will differ between organizations. The underlying principle is:

The strength of the guardrails should reflect the possible harm, autonomy, data sensitivity, reach, and reversibility of the use case.

NIST's AI Risk Management Framework is designed to be adapted to the organization's context, resources, requirements, and risk tolerance rather than applied as one universal checklist.

1. Use Approved Tools and Components

Organizations should maintain a clear view of which AI tools, models, platforms, APIs, extensions, and development frameworks may be used.

Without such guidance, employees often select tools individually based on:

- convenience,
- cost,
- public popularity,
- model performance,
- or recommendations from colleagues.

These criteria may be useful, but they do not show whether the organization has assessed:

- data handling,
- security,

- contractual terms,
- intellectual-property conditions,
- provider dependency,
- logging capabilities,
- support,
- or operational resilience.

The first guardrail is therefore:

Use only approved AI tools and components for organizational work — or request an explicit assessment before introducing a new one.

Approval should not be interpreted as unrestricted permission.

A tool may be approved for:

- non-confidential internal text,
- software-development assistance,
- translation,
- or controlled prototyping,

while remaining prohibited for:

- personal data,
- customer decisions,
- autonomous communication,
- or legally consequential processes.

The approval must therefore connect the tool to a permitted context.

A practical tool entry should state:

Field	**Example**
Tool	Approved enterprise AI assistant
Status	Adopt
Permitted data	Public and approved internal information
Prohibited data	Special-category personal data and classified customer information
Permitted uses	Drafting, summarization, ideation
Restricted uses	Customer communication requires human approval
Prohibited uses	Autonomous personnel or legal decisions
Owner	AI Platform Team
Review trigger	**Contract, model, hosting, or data-policy change**

NIST's Generative AI Profile recommends acceptable-use policies for third-party generative AI systems, documentation of dependencies, fallback arrangements, provider communication procedures, and continuous monitoring of third-party systems in deployment.

Tool approval should consequently include both the technology and the intended use.

An approved tool is not approved for every purpose.

Teams Should Be Able to Request New Tools

An approved-tool policy should not freeze the technology landscape.
Teams need a lightweight mechanism to propose and evaluate new tools.

The assessment can examine:

- intended use,
- required data,
- provider and hosting model,
- security,
- contractual conditions,
- licensing,
- model performance,
- integration,
- monitoring,
- alternatives,
- and exit options.

Depending on the risk, the result may be:

- Adopt,
- Trial,
- Assess,
- Hold,
- or Reject.

This links the guardrail directly to the Tech Radar and Decision Records described in the previous section.

2. Protect Data, Confidential Information, and Intellectual Property

The second guardrail concerns what information may enter the AI system.

AI tools can make it deceptively easy to copy:

- customer information,
- internal documents,
- source code,
- contracts,

- personal data,
- trade secrets,
- or unpublished intellectual property

into an external service.

The user may see only a simple text box. But behind that interface may be a complex chain of providers, models, storage systems, logs, subprocessors, and geographic processing locations.

The practical guardrail should therefore be simple:

Classify the data before using it with AI. Do not enter information into an AI system unless the tool and the intended processing are approved for that data classification.

Teams should know:

- whether the data contains personal information,
- whether it is confidential or contractually protected,
- who owns it,
- whether it may be used for the intended purpose,
- whether it leaves the organization,
- whether the provider retains it,
- whether it may be used for model training,
- and how it can be deleted.

Where possible, early experiments should reduce exposure through:

- anonymized data,
- pseudonymized data,
- synthetic data,
- masked values,
- representative test cases,
- or restricted datasets.

This is not only a privacy concern.

Data protection also covers:

- business confidentiality,
- customer contracts,
- copyright,
- trade secrets,
- intellectual property,
- and security-relevant information.

NIST's Generative AI Profile recommends data-security and privacy controls, policies for third-party data use, documentation of incidents involving external data or systems, and controls tailored to the way foundation models, fine-tuned models, and embedded AI components are used.

Data Minimization Is Also a Risk-Control Mechanism

Teams should ask:

What is the minimum information the system needs to perform this task?

An AI assistant may not need:

- the customer's full identity,
- an entire case history,
- complete documents,
- or production-system access.

Restricting the data reduces:

- privacy exposure,
- security impact,
- incorrect correlations,
- unintended secondary use,
- and the consequences of an incident.

Data access should increase only when the expected value and available evidence justify the additional exposure.

3. Define Permitted Uses, Human Oversight, and Non-Negotiable Boundaries

A tool can be safe for one purpose and unacceptable for another.

The third guardrail therefore concerns the **intended use**.

Every material AI solution should define:

- permitted uses,
- restricted uses,
- prohibited uses,
- required human oversight,
- and escalation conditions.

For example, a generative AI system might be permitted to:

- draft internal text,
- summarize documents,
- suggest alternatives,
- or assist trained specialists.

The same system might be restricted from:

- sending customer messages without approval,
- interpreting contracts as legal advice,
- evaluating employees,
- making commitments on behalf of the organization,
- or executing actions autonomously.

The practical rule is:

Do not expand the use of an AI system beyond its approved purpose without a new risk review.

This addresses one of the most subtle forms of AI risk: gradual purpose expansion.

A system may begin as a drafting assistant. Over time, employees may start copying its answer directly. The assistant may later be connected to a workflow. Eventually, the system may act autonomously even though no explicit decision was made to increase its authority.

Guardrails should prevent this silent transition.

Human Oversight Must Be Meaningful

"Human in the loop" is not a sufficient control description.

The organization should define:

- which outputs must be reviewed,
- who performs the review,
- what the reviewer must check,
- which information is available,
- how much time is expected,
- what can be overridden,
- when escalation is mandatory,
- and who can stop the system.

For high-risk AI systems, the EU AI Act requires designs that enable effective human oversight and help overseers understand system capabilities and limitations, interpret outputs, guard against automation bias, intervene, and stop the system where necessary.

These specific requirements apply to high-risk systems, but the management principle is broadly useful:

Human oversight is effective only when the human has sufficient competence, information, time, and authority to challenge the AI.

Ethical Principles Must Become Observable Behaviors

Principles such as:

- fairness,
- transparency,
- accountability,
- privacy,
- reliability,

- human oversight,
- and sustainability

are important, but too abstract for everyday decisions unless they are translated into concrete expectations.

For example:

Principle	Practical guardrail
Transparency	Inform users when relevant content or recommendations are AI-generated
Accountability	A named person remains responsible for every consequential action
Fairness	Test relevant groups and error distributions rather than only overall performance
Human oversight	People can review, challenge, override, and escalate
Privacy	Do not process unapproved personal or confidential data
Reliability	Define performance boundaries, monitoring, and fallback procedures
Sustainability	Consider usage, infrastructure, and operational cost before scaling

4. Ensure Accountability, Traceability, and Reviewability

The fourth guardrail ensures that the organization can understand how an AI-assisted outcome was produced and who remains responsible for it.

The practical rule is:

Consequential AI-assisted actions must have a named accountable person and sufficient traceability to reconstruct what happened.

Depending on the use case, traceability may include:

- tool and model version,
- system configuration,
- relevant prompt or instructions,
- source data or documents,
- output,
- human edits,
- approval,
- timestamp,
- user,
- and linked Decision Record.

Not every productivity prompt needs permanent logging.

Logging should be proportionate to:

- the consequence of the task,
- legal obligations,
- incident-investigation needs,
- privacy considerations,
- and the level of automation.

A system that drafts an internal headline requires less traceability than a system that recommends whether a customer claim should be accepted.

For high-risk AI systems, Article 12 of the EU AI Act requires automatic logging capabilities that support lifecycle traceability, risk identification, operational monitoring, and post-market monitoring.

This legal requirement does not automatically apply to every AI tool. But it illustrates the broader operating principle:

The greater the consequence of an AI-supported decision, the stronger the evidence trail should be.

Accountability Must Remain Human

Teams should not write:

"The AI decided."

AI may generate, predict, rank, classify, or recommend.

The organization decides:

- whether to use the system,
- which data it receives,
- which process it influences,
- how its output is interpreted,
- and which action follows.

The accountable human role should therefore remain visible even when the technical process is highly automated.

Exceptions Need Documentation

Guardrails should also define how exceptions are handled.

A team may need:

- a new tool,
- broader data access,
- a different model,
- a temporary control,
- or a use outside the current approved boundary.

Exceptions should not be impossible, but they should be explicit.

A lightweight exception record should state:

- what is requested,

- why it is needed,
- which risk changes,
- which temporary controls apply,
- who approved it,
- how long it remains valid,
- and what triggers reassessment.

Otherwise, temporary exceptions gradually become undocumented standard practice.

5. Monitor, Report, and Improve

The final guardrail concerns what happens when reality contradicts the original assumptions.

Users should know how to report:

- incorrect output,
- harmful or biased behavior,
- privacy concerns,
- security vulnerabilities,
- unexpected costs,
- model changes,
- misuse,
- near misses,
- and controls that do not work.

The practical rule is:

Unexpected behavior must be easy to report, and material findings must feed back into risk assessment and governance decisions.

Reporting should not depend on users knowing which department owns the problem.

A central AI incident or feedback channel can route findings to:

- the initiative owner,
- Security,
- Data Protection,
- Legal,
- AI Governance,
- the model owner,
- or operational support.

The organization should define:

- what constitutes an incident,
- which issues require immediate escalation,

- who investigates,
- who may restrict or suspend the system,
- how affected users are informed,
- and how the learning is shared across initiatives.

The joint secure-AI-development guidelines published by the UK National Cyber Security Centre and international partners recommend lifecycle security measures including threat modelling, protection of infrastructure and models, incident-management procedures, logging and monitoring, secure updates, and sharing lessons learned.

NIST's Generative AI Profile similarly recommends continuous monitoring of third-party systems, documentation of incidents, fallback arrangements, and updated policies as systems and dependencies change.

Near Misses Matter

Organizations should not wait for visible harm.

A near miss may reveal:

- a control that nearly failed,
- an output that was corrected only by chance,
- misuse that had no immediate consequence,
- or a model change that was detected before broader deployment.

Near misses are valuable evidence because they expose weaknesses before the full impact occurs.

The objective should not be to punish every error.

It should be to make risk visible early enough to improve the system.

Monitoring Must Trigger Decisions

A dashboard alone does not manage risk.

Monitoring becomes useful only when thresholds are connected to actions.

For example:

Indicator	Trigger	Action
Critical output error	One confirmed case	Suspend affected function and investigate
Error rate	Above approved threshold	Reassess model and control effectiveness
Model update	Major provider release	Repeat defined evaluation tests
Usage expansion	New department or use case	Conduct new risk review
Cost increase	Above business-case tolerance	Reassess viability
Oversight failure	Review skipped or ineffective	Restrict use and redesign the control
Security event	Confirmed or suspected compromise	Activate incident-response process

This closes the feedback loop between guardrails, monitoring, Decision Records, and stage-gate governance.

Make the Safe Path the Easy Path

Guardrails fail when following them is substantially harder than bypassing them.

Employees will seek alternatives when:

- approved tools are unavailable,
- the approval process takes months,
- policies are difficult to understand,
- data rules are ambiguous,
- or no support exists for legitimate experiments.

A mature guardrail system should therefore provide:

- approved tools that meet common needs,
- clear examples of allowed and prohibited use,
- reusable technical components,
- data-classification guidance,
- fast consultation channels,
- standard experiment environments,
- risk-assessment templates,
- and transparent exception processes.

The secure-AI-development guidelines explicitly recommend making it easy for users to do the right thing and treating security as a lifecycle requirement rather than an afterthought.

This leads to an important design principle:

A guardrail should reduce decision friction for safe work — not merely increase friction for risky work.

Guardrails Need AI Literacy

Policies alone cannot anticipate every situation.

Employees need enough AI literacy to recognize when a task may involve:

- sensitive data,
- unreliable output,
- inappropriate reliance,
- legal or ethical impact,
- security concerns,
- or a use outside the approved boundaries.

The EU AI Act requires providers and deployers of AI systems to take measures, to their best extent, to ensure a sufficient level of AI literacy among relevant staff and other people operating AI systems on their behalf, taking account of their knowledge, experience, training, education, and the context of use.

AI literacy should therefore go beyond prompt-writing techniques.

People need to understand:

- that fluent output can still be wrong,
- when sources need verification,
- how automation bias works,
- which data may be entered,
- who remains accountable,
- where the approved boundaries lie,
- and how to report unexpected behavior.

Training should also be role-specific.

A casual user, software engineer, model owner, human overseer, and executive sponsor do not need the same depth of knowledge.

A Practical Guardrail Card

Each organization-wide guardrail can be documented in a compact format:

Field	Question
Guardrail	What boundary must be respected?
Purpose	Which risk does it reduce?
Applies to	Which users, systems, data, or phases are covered?
Allowed	What may teams do independently?
Conditional	What requires defined controls or approval?
Prohibited	What must not be done?
Evidence	How can compliance and effectiveness be demonstrated?
Owner	Who maintains and interprets the guardrail?
Exception path	How can a justified exception be requested?
Review trigger	When must the guardrail be updated?

Example:

Field	Example
Guardrail	No autonomous external communication
Purpose	Prevent incorrect or unauthorized commitments
Applies to	Generative AI used in customer-facing processes
Allowed	Drafting responses for trained employees
Conditional	Sending after documented human approval
Prohibited	Direct autonomous sending
Evidence	Approval log and sampled quality reviews
Owner	Head of Customer Operations
Exception path	AI Steering Group approval
Review trigger	Proposed increase in autonomy or serious incident

This makes the guardrail operational, testable, and reviewable.

Guardrails Create Freedom Within Boundaries

Good guardrails do not eliminate judgment.

They clarify where teams may exercise judgment independently and where organizational accountability is required.

They allow teams to:

- explore ideas,
- build prototypes,
- test models,
- and learn quickly

without repeatedly negotiating the same foundational questions.

At the same time, they prevent experimentation from silently becoming unrestricted operational use.

The practical balance is:

Freedom to innovate within clear ethical, technical, and operational boundaries.

This reflects the central operating principle of the handbook: approved tools, protected data, ethical and compliant use, clear accountability, and continuous improvement provide the operational foundation for responsible AI adoption.

However, even well-designed frameworks can fail when organizations turn them into bureaucracy, treat compliance as a substitute for judgment, or confuse documentation with actual risk reduction.

Common Risk Management Anti-Patterns

1. Risk Management Happens Only Once

The team performs a detailed risk assessment during scoping.

Risks are identified, evaluated, assigned a score, and entered into a register. The initiative then receives approval, and the assessment remains largely unchanged while the project moves through prototyping, piloting, rollout, and operations.

This creates a false sense of stability.

The original assessment was based on assumptions about:

- the intended use,
- the available data,
- the selected model,
- the provider,
- the number of users,
- the operating environment,
- and the effectiveness of the planned controls.

Once these assumptions change, the assessment may no longer describe the actual system.

NIST states that AI risk management should be continuous, timely, and performed throughout the AI lifecycle. For high-risk AI systems, Article 9 of the EU AI Act similarly defines risk management as a continuous iterative process requiring regular review and updating.

Typical symptoms

- The risk register still refers to the prototype architecture.
- New model versions are introduced without reassessment.
- A pilot is expanded to new users without reviewing the context.
- Risk reviews happen only before formal steering meetings.
- Operational incidents do not change the original risk rating.

Better approach

Treat the risk assessment as a living representation of the current system.

Update it when:

- the use case changes,
- new data is introduced,
- the model or provider changes,
- usage expands,
- autonomy increases,
- an incident or near miss occurs,

- or monitoring shows that assumptions or controls no longer hold.

A completed risk assessment is not the objective. A current risk assessment is.

2. The Framework Becomes a Checklist

The organization adopts a respected framework and turns every category into a mandatory checkbox.

Teams are asked whether they have considered:

- fairness,
- transparency,
- privacy,
- explainability,
- security,
- human oversight,
- and accountability.

Every answer eventually becomes “yes.”

The project appears compliant, but the checklist does not reveal:

- which risks are material,
- which trade-offs exist,
- what evidence supports the answer,
- or whether the remaining risk is acceptable.

NIST explicitly states that the AI RMF actions do not constitute a checklist or necessarily follow a fixed order. The Playbook is intended to be tailored to the organization, system, resources, and context.

Typical symptoms

- Every initiative completes the same form.
- Low-impact experiments and consequential systems face identical reviews.
- Teams optimize for answering the questions rather than understanding the risk.
- All categories receive a green status without a discussion of trade-offs.
  Completion of the template is treated as proof of trustworthiness.

Better approach

Use frameworks to guide thinking, not replace it.

Ask:

- Which risk domains are most material for this use case?
- What could cause serious harm or invalidate the business case?
- What is highly uncertain?
- Which evidence would change the decision?
- Which controls need to be tested rather than merely documented?

A framework should improve judgment. It should not automate judgment.

3. Documentation Is Confused with Evidence

The team presents:

- a policy,
- a test plan,
- a model card,
- a training concept,
- a human-review procedure,
- and an incident-management document.

All required artifacts exist.

But their existence does not prove that the system or its controls work.

A testing strategy is not a test result.

A human-review procedure is not evidence that reviewers detect errors.

A training document is not evidence that users understand the system.

A monitoring dashboard is not evidence that someone acts when a threshold is exceeded.

NIST distinguishes between documented governance processes, risk measurement, and explicit go/no-go decisions. It also encourages organizations to evaluate whether their policies, practices, indicators, and measurements actually improve their ability to manage risk.

Typical symptoms

- Gate presentations focus on completed deliverables.
- Controls are approved before they have been tested.
- A policy is used as proof that user behavior is controlled.
- The team shows a dashboard but cannot explain the thresholds.
- Reviewers ask, “Do we have the document?” rather than, “What did we learn?”

Better approach

Connect every important claim to evidence.

Claim	Activity	Evidence
Users will adopt the system	Conduct pilot training	Observed usage and task-completion data
Human review prevents harmful output	Define review procedure	Measured detection rate under realistic workload
The model is sufficiently reliable	Run evaluation tests	Error distribution across relevant cases
The solution is economically viable	Prepare cost model	Actual pilot costs and scaled usage projections
Monitoring controls drift	Build dashboard	Demonstrated detection and tested response

Claim	Activity	Evidence
		process

Documents describe the intended system. Evidence shows how the real system behaves.

4. The Model Is Evaluated in Isolation

The team focuses heavily on model performance.

It measures:

- accuracy,
- precision,
- recall,
- hallucination rates,
- or benchmark scores.

These metrics matter, but they describe only one component of the AI-enabled system.

AI risks emerge from the interaction between:

- the model,
- the data,
- the interface,
- the user,
- the surrounding process,
- organizational incentives,
- and the consequences of the resulting action.

NIST describes AI systems as inherently socio-technical. Their risks and benefits emerge from the interplay of technical characteristics, human behavior, operators, other systems, and the deployment context.

A model may perform well during evaluation and still fail because:

- users misunderstand the output,
- the wrong source documents are retrieved,
- human reviewers approve everything automatically,
- the workflow encourages inappropriate reliance,
- or the system is used outside the tested scope.

Typical symptoms

- The main success criterion is one average model score.
- User behavior is treated as a change-management issue for later.
- The evaluation does not include the actual process.
- The team ignores the cost and effectiveness of human review.
- Edge cases with serious consequences are hidden by strong average performance.

Better approach

Evaluate the complete system under realistic conditions.

Measure:

- technical performance,
- error types,
- user behavior,
- human-review effectiveness,
- process outcomes,
- operating costs,
- affected groups,
- and the consequences of undetected errors.

The threshold should depend on the intended use.

A model is not sufficiently reliable in the abstract. It is sufficiently reliable—or not—for a specific purpose, context, and set of controls.

5. “Human in the Loop” Is Treated as a Universal Safeguard

The project identifies a serious risk and adds:

“A human will review the output.”

The risk is then marked as mitigated.

But human oversight can fail when reviewers:

- lack relevant expertise,
- have insufficient time,
- cannot detect plausible errors,
- do not receive the necessary context,
- are encouraged to follow the AI recommendation,
- or lack authority to override or stop the system.

Article 14 of the EU AI Act requires, for high-risk AI systems, that oversight measures enable people to understand system capabilities and limitations, recognize potential over-reliance, interpret outputs, override or reverse them, and intervene or stop the system.

The legal obligation applies specifically to high-risk AI systems, but the management lesson is broader:

Placing a person in the process does not automatically create effective control.

Typical symptoms

- The control description contains only the words “human review.”
- No one has defined what must be checked.
- Reviewers approve nearly every output.
- Review workload increases but staffing does not.
- The reviewer remains accountable but cannot change or stop the system.

- No one measures whether the review catches important errors.
  Better approach

Define human oversight operationally:

- Who reviews?
- What competence is required?
- Which outputs must be reviewed?
- Which information must be available?
- What is the expected workload?
- Which errors must trigger escalation?
- Can the reviewer reject, reverse, or stop the outcome?
- How will control effectiveness be measured?

Human oversight is a control only when the human can meaningfully influence the outcome.

6. Everybody Owns the Risk

AI initiatives often involve many functions:

- Business,
- Product,
- Data Science,
- IT,
- Security,
- Privacy,
- Legal,
- HR,
- Operations,
- and executive sponsors.

This is necessary because AI risk is multidisciplinary.

But involvement is not the same as ownership.

When the risk is assigned to “the project,” “the steering group,” or “all stakeholders,” no individual may feel responsible for:

- maintaining the assessment,
- implementing controls,
- collecting evidence,
- escalating threshold breaches,
- or recommending that the system be restricted.

NIST states that effective AI risk management requires appropriate accountability mechanisms, explicit roles and responsibilities, organizational commitment, and suitable incentive structures.

Typical symptoms

- Risks are assigned to departments rather than people or roles.
- Legal assumes the business accepted the risk.
- The business assumes Security approved the system.
- The steering group discusses the risk but does not assign actions.
- No one knows who can suspend the system.

Better approach

Separate:

- the **risk owner,**
- the **control owner,**
- the **evidence owner,**
- and the **risk-acceptance authority.**

For every material risk, document:

Question	Required answer
Who manages the risk?	Named risk owner
Who operates the safeguards?	Named control owner or owners
Who validates the evidence?	Named evidence owner
Who accepts the residual risk?	Authorized decision-maker
Who can stop the system?	Explicit stop authority
When must the risk be reviewed?	Defined trigger

Collective responsibility requires explicit individual accountability.

7. Every AI Initiative Receives the Same Governance

To appear consistent, the organization creates one approval process for every AI use case.

An employee experimenting with non-confidential text must follow the same process as a system that:

- processes personal data,
- influences employment,
- communicates autonomously with customers,
- or supports consequential decisions.

This creates two problems.

Low-risk teams experience unnecessary friction and begin bypassing governance.

High-risk teams may satisfy the standard process even though their use case requires much stronger evidence and controls.

NIST's framework is intended to be tailored to the organization's resources, capabilities, context, and risk tolerance. The EU AI Act likewise applies a risk-based structure, with stronger requirements for defined high-risk systems.

Typical symptoms

- Every experiment requires the same committee.
- Governance effort depends on project size rather than potential harm.
- Low-impact use cases wait months for approval.
- High-impact use cases complete a generic checklist.
- Teams cannot explain how the governance level was determined.

Better approach
Scale governance according to:

- potential impact,
- data sensitivity,
- number of affected people,
- level of autonomy,
- reach,
- reversibility of harm,
- regulatory relevance,
- and uncertainty.

A low-exposure experiment can use lightweight guardrails.

A high-impact system requires formal evidence, independent review, clear risk acceptance, and continuous monitoring.

Proportional governance protects both safety and speed.

8. Stage Gates Become Status Meetings

The initiative team presents:

- the completed milestones,
- the current schedule,
- the budget,
- the latest prototype,
- and the activities planned for the next phase.

The committee asks questions but avoids a clear decision.
The project then continues because stopping it would be uncomfortable. This weakens the central purpose of the gate.

Official Stage-Gate guidance describes gates as forward-looking decisions about continued investment, resources, priorities, and the information required to address remaining gaps.

Typical symptoms

- Every initiative receives a “Go.”
- Gatekeepers attend but lack decision authority.
- Evidence is distributed too late for meaningful review.
- The meeting ends with vague requests for improvement.
- No one states which exposure has been approved.
- Known risks are carried forward without explicit acceptance.

Better approach
Every gate should conclude with a documented decision:

- **Go**
- **Rework**
- **Restrict**
- **Pause**
- **Kill**

The decision should state:

- what is approved,
- which scope and users are included,
- which data may be used,
- which controls are mandatory,
- which residual risks were accepted,
- who accepted them,
- and what triggers a new review.

A gate without a clear decision is a presentation, not a control mechanism.

9. Monitoring Exists Without Intervention

The production system has:

- logs,
- quality metrics,
- cost dashboards,
- usage statistics,
- drift indicators,
- and incident channels.

But no one has defined what should happen when the values change.

The organization observes the risk without managing it.

NIST recommends continuous monitoring for performance degradation, unusual behavior, near misses, attacks, and changing impacts, including monitoring of third-party and pre-trained components. For high-risk systems, Article 72 of the EU AI Act requires active and systematic post-market collection and analysis of relevant performance data throughout the system lifetime.

Typical symptoms

- Thresholds are not defined.
- Alerts are repeatedly ignored.
- No one owns the response.
- Provider updates do not trigger reevaluation.
- Near misses are logged but not discussed.
- The system continues operating while an investigation remains open.

Better approach
Connect indicators to decisions.

Signal	Trigger	Required action
Critical harmful output	One confirmed event	Restrict or suspend affected function
Error rate	Above approved threshold	Reassess model and controls
Provider model update	Material version change	Repeat defined evaluation suite
Drift	Outside accepted range	Investigate data and model behavior
Cost	Above viability threshold	Review the business case
Oversight failure	Control repeatedly bypassed	Redesign or restrict the process
New use case	Scope expansion	Perform new risk review

Monitoring manages risk only when observations trigger action.

10. Teams Are Rewarded for Proving the Project Right

An initiative receives executive attention, budget, and public support.

The team gradually feels responsible for demonstrating success rather than investigating whether the assumptions remain valid.

Negative findings are reframed as temporary issues.
Failed experiments are repeated until they produce a positive result.
Risks are softened before steering meetings.
The project survives, but the learning system collapses.

NIST's framework emphasizes a culture that prioritizes identification of risks, multidisciplinary perspectives, open sharing of assumptions, and explicit go/no-go decisions. This implies that

organizations need conditions in which uncomfortable evidence can be surfaced without being treated automatically as failure.

Typical symptoms

- Every experiment is presented as successful.
- Teams hide results that could threaten funding.
- “Kill” decisions are treated as personal or organizational failure.
- Sponsors defend the project instead of evaluating the evidence.
- Risks disappear from presentations as the project gains visibility.
- The team optimizes the demonstration rather than the decision.

Better approach

Reward teams for improving decision quality.

A valuable experiment may show that:

- the problem is less important than expected,
- AI does not outperform a simpler solution,
- required human review eliminates the business benefit,
- the data is unsuitable,
- the risk is disproportionate,
- or the use case should be narrowed.

Stopping such an initiative early protects resources and organizational trust.

The purpose of experimentation is not to prove that the project should continue. It is to discover whether it should.

A Compact Anti-Pattern Diagnostic

Before approving the next phase, ask:

Diagnostic question	Anti-pattern it exposes
When was the risk assessment last updated?	One-time risk assessment
Which risks matter most in this context?	Checklist governance
What evidence proves that the control works?	Documentation without evidence
Have we evaluated the complete work system?	Model-only evaluation
Can the human reviewer really intervene?	Symbolic human oversight
Who owns and who accepts each residual risk?	Collective but unclear ownership
Is governance proportional to potential impact?	One-size-fits-all governance
What exact decision is being made at the gate?	Status-meeting gates
Which signals trigger restriction or suspension?	Monitoring without intervention

Diagnostic question	Anti-pattern it exposes
Which result would cause us to stop?	Confirmation-driven experimentation

This diagnostic can be used:

- before a gate meeting,
- during an operational review,
- when designing an AI governance process,
- or when an initiative appears formally compliant but still feels uncontrolled.

The Deeper Pattern Behind the Anti-Patterns

Most of these failures have the same root cause:

The organization separates governance from the real work of learning and deciding.

Risk management becomes paperwork.
Human oversight becomes a sentence in a policy.
Monitoring becomes a dashboard.
Stage gates become presentations.
Governance becomes a committee.

The alternative is a connected operating system:

- experiments generate evidence,
- evidence changes the risk assessment,
- controls are tested,
- risks have named owners,
- gate decisions define permitted exposure,
- Decision Records preserve the reasoning,
- guardrails guide everyday work,
- and monitoring reopens decisions when conditions change.

This is the practical operating model developed throughout this handbook: move quickly through small learning cycles, but increase investment and exposure only through explicit, evidence-based risk decisions.

Risk management should not slow learning down.

It should prevent the organization from scaling assumptions that have not yet earned its trust.

Final Thoughts: Move Fast, but Responsibly

AI projects do not need less ambition.

They need better mechanisms for deciding how much uncertainty, investment, and organizational exposure the company is prepared to accept.

Conventional IT risk management remains necessary. AI systems still need secure architecture, reliable infrastructure, clear responsibilities, realistic budgets, and disciplined project execution.

But these controls are no longer sufficient on their own.

AI projects introduce additional uncertainty around data quality, model behavior, provider dependency, drift, user interaction, human oversight, adoption, ethics, regulation, and trust. These risks do not remain stable. They evolve as the initiative moves from an idea to a prototype, from a prototype to a real-world pilot, and from a pilot into broad operational use.

This is why AI risk management cannot be reduced to an assessment performed before development begins.

It must become a continuous learning and decision system.

The Goal Is Not to Eliminate Every Risk

An organization that attempts to eliminate every uncertainty before experimenting will move too slowly to learn.

Many of the most important questions cannot be answered through planning alone:

- Will users actually benefit from the solution?
- Is the available data suitable?
- How will the model behave outside controlled test cases?
- Will human reviewers detect plausible but incorrect output?
- Can the solution be operated economically at scale?
- Will the controls remain effective under real working conditions?

The organization needs experiments to answer these questions.

At the same time, experimentation should not become an excuse for uncontrolled exposure.

A prototype does not automatically justify access to production data. A successful demonstration does not prove that a system is ready for real users. A promising pilot does not demonstrate that controls, costs, and model performance will remain acceptable at scale.

The goal of risk management is therefore not zero risk.

The goal is to:

- make uncertainty visible,
- test the most critical assumptions,
- limit the potential impact of failure,
- reduce risks where possible,
- accept residual risks consciously,

- and increase exposure only when the available evidence justifies it.

Risk management should not prevent experimentation. It should prevent organizations from scaling assumptions that have not yet earned their trust.

Be Agile Within the Stages and Evidence-Driven at the Gates

The model developed in this handbook combines two complementary mechanisms.

Inside each phase, teams work iteratively.

They build small increments, test assumptions, collect evidence, identify emerging risks, and adapt the model, data, process, controls, or use case.

At the stage gates, the organization makes an explicit decision.

It determines whether the accumulated evidence justifies:

- further investment,
- access to more sensitive data,
- involvement of real users,
- deeper integration into business processes,
- increased autonomy,
- or broader operational use.

This distinction allows both speed and control:

Agile iterations manage learning. Stage gates manage exposure.

The stage-gate process should not protect the original plan.

It should protect the organization while allowing the initiative to evolve.

A team may discover that the original scope is too broad. It may reduce autonomy, change from automation to augmentation, restrict the user group, select another provider, or replace the AI approach with a simpler solution.

These are not signs that the process failed.

They are signs that the learning system is working.

Risk Management Requires More Than a Risk Register

A risk register can document current concerns, but it cannot manage them on its own.

Effective AI risk management requires a connected operating model:

- The **six risk domains** ensure that value, business viability, data, models, operations, compliance, ethics, and trust are considered together.
- **Iterative experiments** convert uncertainty into evidence.
- **Stage gates** determine whether the next level of exposure is justified.
- **Risk owners and decision authorities** make accountability explicit.
- **Decision Records** preserve the reasoning, accepted risks, restrictions, and review triggers.
- The **Tech Radar** distributes project-specific learning across the organization.

- **Guardrails** translate governance into practical boundaries for everyday work.
- **Monitoring** reveals when assumptions, controls, providers, data, or operating conditions have changed.
- **Reassessment** ensures that approval is not treated as permanent.

These mechanisms should reinforce one another.

Monitoring should trigger risk reviews. Risk reviews should influence stage-gate decisions. Important decisions should create Decision Records. Project evidence should update the Tech Radar and organizational guardrails. New initiatives should benefit from what earlier projects learned.

This turns isolated AI projects into an organizational learning system.

Good Governance Should Increase the Ability to Act

Governance is often perceived as an obstacle because many governance systems focus primarily on approval, documentation, and control.
But governance that only slows teams down is poorly designed.
Good governance creates clarity.

Teams know:

- which tools they may use,
- which data is permitted,
- which experiments they can run independently,
- which controls are mandatory,
- which decisions require escalation,
- and where the non-negotiable boundaries lie.

Within these boundaries, teams should have the freedom to explore, test, and adapt.

The strength of the governance should increase with:

- the sensitivity of the data,
- the reach of the system,
- the level of autonomy,
- the consequences of an incorrect output,
- the difficulty of reversing harm,
- and the uncertainty surrounding the use case.

The objective is not maximum governance.

It is the appropriate level of governance for the current exposure.

Strong guardrails should make safe experimentation easier—not make all experimentation harder.

Trust Must Be Earned and Maintained

Trust is not created by declaring an AI system trustworthy.

It emerges when people can understand:

- what the system is intended to do,
- where its boundaries lie,
- how its performance was evaluated,
- which risks remain,
- who is accountable,
- how decisions can be challenged,
- and what happens when something goes wrong.

Trust also does not mean unquestioned reliance.

Users should trust the system enough to use it, but remain sufficiently critical to recognize its limitations. They need the competence, time, information, and authority to intervene when the system produces an inappropriate result.

This requires transparency, effective human oversight, clear accountability, and visible responses to incidents and feedback.

Trust can take a long time to build and be lost through a single serious incident.

Risk management therefore protects more than budgets, systems, and regulatory compliance.

It protects the organization's ability to use AI at all.

Three Principles to Keep

When designing an AI risk-management system, three principles are particularly important.

Start small enough to fail safely.

Early experiments should create meaningful evidence without exposing the organization to the full impact of an unfinished solution.

Scale evidence, not confidence.

A persuasive presentation, strong sponsorship, or impressive demonstration should not substitute for measured results under realistic conditions.

Keep decisions reviewable and reversible.

Every approval should define its scope, conditions, residual risks, owners, and review triggers. When the context changes, the decision must be reconsidered.

These principles create a practical balance between innovation and responsibility.

Move Fast—But Responsibly

Organizations do not need to choose between speed and safety.

They need an operating model that uses speed to generate learning and structure to determine how that learning should influence investment, permissions, and risk exposure.

Move fast enough to learn before the opportunity disappears.

Move carefully enough to recognize what the evidence is actually telling you.
And scale only what has earned the right to scale.

The final message from the underlying model can therefore remain simple:

Innovation × Trust × Compliance = Sustainable AI Success

The multiplication sign is deliberate.
Innovation without trust will not be adopted.
Trust without responsible controls will not last.
Compliance without real value will produce safe systems that nobody needs.

Sustainable AI success requires all three.

Move fast—but responsibly.

About the Author

David Theil works at the intersection of AI strategy, organizational change, agile product development, and software engineering. His professional background includes roles as Lead Organizational Developer, Agile Coach, Scrum Master, Product Owner, and Head of Software Engineering, where he led an organization of 40 people.

Since 2011, he has published practical articles and books on agile software development, lean product management, Scrum, digital transformation, and software engineering. His work focuses on helping teams move beyond mechanical process adoption and feature-factory behavior toward evidence-driven product development.

This handbook extends that perspective to AI initiatives: teams need freedom to learn, but organizations also need explicit mechanisms for risk ownership, evidence, traceability, and responsible decisions.

Selected author platforms

- DigitalisierungsCoach
- LinkedIn
- Medium

About DigitalisierungsCoach

DigitalisierungsCoach is David Theil's publication platform for practical guidance on digital products, agile product development, Scrum, software engineering, digitalization, and organizational transformation.

Legal and Professional Disclaimer

This handbook is provided for general informational and educational purposes. It presents an organizational management model and does not constitute legal, regulatory, compliance, cybersecurity, financial, employment, or other professional advice.

- The classification of an AI system and the obligations that apply to providers, deployers, importers, distributors, product manufacturers, or other actors depend on the facts, intended purpose, jurisdiction, sector, and legal role.
- References to the EU Artificial Intelligence Act, GDPR, ISO standards, NIST guidance, or other frameworks do not replace a qualified assessment of the requirements applicable to a specific organization or system.
- ISO standards are copyrighted publications. This handbook refers to their public titles and general scope; it does not reproduce or replace the standards.
- Technology providers may change models, contracts, pricing, data-processing conditions, and technical behavior. Current provider documentation and contractual terms should be reviewed.
- The author makes no warranty that the handbook is complete, error-free, or suitable for every context. Decisions should be made with appropriate specialist advice and organizational authority.

Use this handbook to improve questions, evidence, and decisions — not to replace qualified professional judgment.

Sources and Further Reading

Primary frameworks, standards, regulations, and methods used in the handbook

Inline hyperlinks in the handbook point readers to the relevant official or primary source where possible. The following bibliography provides the full source set used to develop the handbook's risk-management, governance, lifecycle, evidence, traceability, and guardrail concepts.

All online sources were accessed on 16 July 2026. Readers should verify whether newer editions, revisions, implementation guidance, or consolidated legal texts are available.

AI risk management and governance

National Institute of Standards and Technology (NIST). Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1, January 2023. DOI: 10.6028/NIST.AI.100-1. *Foundational reference for lifecycle-oriented, socio-technical AI risk management and the Govern, Map, Measure, and Manage functions.*

National Institute of Standards and Technology (NIST). NIST AI RMF Playbook. Complete online companion to AI RMF 1.0, first complete version released March 2023. *Provides adaptable actions and documentation practices; it is not intended as a universal checklist.*

National Institute of Standards and Technology (NIST). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1, July 2024. DOI: 10.6028/NIST.AI.600-1. *Companion profile addressing risks and controls that are unique to or intensified by generative AI.*

Regulation and international standards

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. Adopted 13 June 2024; commonly referred to as the European Union Artificial Intelligence Act. *Primary legal source for the handbook's references to AI literacy, risk management, technical documentation, logging, human oversight, deployer obligations, and post-market monitoring.*

International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 23894:2023 — Information technology — Artificial intelligence — Guidance on risk management. Edition 1, February 2023. *Guidance for integrating AI-specific risk management into organizational activities and functions.*

International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 42001:2023 — Information technology — Artificial intelligence — Management system. Edition 1, December 2023. *Requirements for establishing, implementing, maintaining, and continually improving an AI management system.*

Agile learning, staged investment, and decision management

Beck, Kent, et al.. Manifesto for Agile Software Development. Published 2001. *Source for the principles of iterative delivery, collaboration, feedback, adaptation, and responding to change.*

Stage-Gate International. The Stage-Gate Model: An Overview. Official overview of stages, decision gates, evidence, resource commitments, and adaptive in-stage work. *Used as the basis for treating gates as forward-looking investment and exposure decisions rather than status meetings.*

Amazon Web Services. Using architectural decision records to streamline technical decision-making for a software development project. AWS Prescriptive Guidance, Darius Kunce and Dominik Goby, March 2022. *Basis for extending Architecture Decision Record practices to AI, risk, and governance decisions.*

Thoughtworks. Technology Radar. Ongoing knowledge-sharing framework for communicating organizational positions on technologies and practices. *Basis for the handbook's Adopt, Trial, Assess, and Hold/Caution model for shared organizational learning and risk appetite.*

Secure development and operational practice

UK National Cyber Security Centre and international partners. Guidelines for Secure AI System Development. Version 1.0, published 27 November 2023. *Lifecycle guidance covering secure design, development, deployment, operation, monitoring, incident management, updates, and information sharing.*

Author's foundational work

Theil, David. *Building Trust — Managing Risk with Agility and Structure in AI Projects*. Presentation and workshop material, 2025. *The original presentation provided the foundation for the six risk domains, the combination of agile learning and staged risk decisions, governance roles, traceability, and practical guardrails developed in this handbook.*

Author profile sources

DigitalisierungsCoach. David Theil — Author profile. Author biography and publication archive.

Medium. David Theil — About. Author biography, professional background, and selected publications.

LinkedIn. David Theil — Professional profile. Professional profile and current career information.

Innovation × Trust × Compliance = Sustainable AI Success

MOVE FAST — BUT RESPONSIBLY

AI projects move fast — and the risks move with them. Traditional risk management was not designed for systems that learn, adapt, and influence decisions. This handbook provides a practical operating model that combines agile learning with structured risk decisions so you can innovate with confidence and scale only what has earned your trust.

Inside, you'll find clear frameworks, decision tools, and real-world guidance to manage uncertainty, protect people, comply with regulations, and create sustainable value.

FOR LEADERS. FOR PRACTITIONERS. FOR ORGANIZATIONS BUILDING THE FUTURE.

Whether you are sponsoring, building, governing, or operating AI systems, this handbook helps you make better decisions at every stage.

WHAT YOU'LL GET

Six connected risk domains

A comprehensive view of AI risk across value, data, models, operations, ethics, compliance, and trust.

Agile learning inside the stages

Iterative experiments that turn uncertainty into evidence and improve decisions.

Stage gates at key decisions

Five possible decisions to manage increasing exposure and investment.

Governance and accountability

Clear roles, ownership, and decision authority across the organization.

Traceable decisions

Decision Records that capture context, evidence, risks, conditions, and review triggers.

Tech Radar and guardrails

Practical tools to steer the use of AI technologies and set clear operating boundaries.

> Risk management should not prevent experimentation. It should prevent organizations from scaling assumptions that have not yet earned their trust.

- ✓ Make uncertainty visible
- ✓ Test the most important assumptions
- ✓ Limit the impact of failure
- ✓ Reduce risks where possible
- ✓ Consciously accept the residual exposure

ABOUT THE AUTHOR

David Theil is an AI strategist and organizational development expert helping organizations design responsible AI strategies, build effective governance, and manage risks in a way that enables innovation. He combines practical experience in digital transformation, agile methods, and risk management with a focus on human impact, change management, and sustainable value creation.

Connect on LinkedIn: linkedin.com/in/davidtheil

- Practical frameworks instead of theory
- Designed for real-world AI initiatives
- Actionable tools you can apply today
- Built for leaders, practitioners, and teams

GET THE COMPLETE HANDBOOK

Download the full 138-page PDF and practical templates

digitalisierungscoach.

digitalisierungscoach.com/ai-risk-management-handbook