

AI Governance & Model Risk Management Framework Guide

Govern / Validate / Monitor / Evidence

A practical enterprise guide to MRM, model validation, risk classification, documentation, and AI-specific controls.

2026 Edition

For model validators, AI governance professionals, and risk leaders.

Dr. Nabanita Sinha

What AI governance is

AI governance is the enterprise system of decision rights, policies, roles, controls, evidence, and oversight used to guide AI from idea through retirement. It connects responsible-AI principles and regulatory obligations to the way teams actually build, buy, validate, approve, operate, change, and retire AI systems.

Why it matters

Decisions affect people

AI can influence credit, employment, pricing, security, healthcare, access, and customer treatment. Governance makes impact and accountability visible.

Models can be wrong or misused

A technically sound model can still create harm when data, assumptions, implementation, users, thresholds, or business processes do not match its intended use.

AI changes after launch

Data, prompts, retrieval corpora, tools, vendors, and user behavior evolve. A one-time approval does not control a changing system.

Evidence protects trust

Inventories, validation reports, approvals, logs, monitoring, issues, and change records let leaders, auditors, and supervisors assess what happened and why.

The outcome

Good governance does not mean zero risk or a large committee for every use case. It means risk-proportionate decisions, clear ownership, independent challenge where needed, usable controls, and evidence that the system remains fit for its intended purpose.

A practical definition

Governance is the operating system for accountable AI decisions. MRM is one discipline inside that system, focused on the risk that models are incorrect, unstable, poorly understood, or used inappropriately.

Model validator and AI governance roles

Model validator

Provides independent and effective challenge. Reviews intended use, methodology, data, assumptions, implementation, testing, performance, limitations, controls, monitoring, and documentation. Produces findings and an opinion on fitness for purpose.

AI governance professional

Designs and operates the enterprise framework: inventory, risk classification, policy, lifecycle gates, committee or approval processes, control mapping, evidence standards, issue management, regulatory alignment, training, reporting, and portfolio oversight.

Model owner / business owner

Owns the business decision, use, benefits, outcomes, resources, compliance, limitations, controls, and residual risk. Ownership cannot be transferred to the developer, validator, vendor, or committee.

Do these jobs require hands-on coding?

Not always. A governance role may not require production coding. A validator may use SQL, Python, statistical tools, or testing platforms depending on the model and organisation. Both roles require technical depth even when coding is not the primary job.

- Understand how data, features, prompts, retrieval, models, tools, thresholds, and human decisions connect.
- Challenge metrics, samples, assumptions, baselines, test design, implementation, and monitoring evidence.
- Read architecture diagrams, model documentation, evaluation reports, logs, issue records, and vendor evidence.
- Explain technical limitations and residual risk to business, audit, risk, legal, and supervisory audiences.

The interview signal is not: Can you train the model yourself? It is: Can you identify what evidence is required, judge whether it is credible, ask the right challenge questions, and make risk visible?

Understanding Model Risk Management

Model Risk Management (MRM) is the coordinated set of practices used to identify, assess, control, validate, monitor, and report the risk created by models. Traditional banking MRM grew around quantitative models, but its core disciplines are highly relevant to AI when extended to the full system and use context.

What is model risk?

Model risk is the potential for adverse consequences from decisions based on incorrect, unstable, poorly implemented, misunderstood, or misused model outputs. Risk can arise from the model itself and from how people, data, systems, and processes use it.

The enterprise MRM lifecycle

- 1 Identify**
Define what counts as a model or AI system and register it before material use.
- 2 Classify**
Assess inherent risk, materiality, complexity, autonomy, exposure, and regulatory relevance.
- 3 Develop or procure**
Set standards for data, design, testing, third-party review, and documentation.
- 4 Validate**
Apply independent, risk-proportionate challenge before use and throughout the lifecycle.
- 5 Approve**
Record decision rights, conditions, limitations, exceptions, and residual-risk acceptance.
- 6 Implement**
Verify that production logic, data, controls, access, and user processes match the approved design.
- 7 Monitor**
Track performance, outcomes, drift, fairness, incidents, overrides, control health, and usage.
- 8 Change**
Assess materiality, re-test, re-validate when required, approve, document, and preserve rollback.
- 9 Remediate**
Prioritise findings, assign owners and dates, apply compensating controls, and escalate overdue issues.
- 10 Retire**
Stop use safely, notify users, archive evidence, manage dependencies, and retain data appropriately.

SR 11-7, SR 26-2, and SS1/23

SR 11-7 (United States, 2011)

The foundational Federal Reserve and OCC guidance many interviewers still reference. It organised sound MRM around robust development, implementation and use; rigorous validation; and governance, policies, and controls. Its classic validation vocabulary includes conceptual soundness, ongoing monitoring, and outcomes analysis.

SR 26-2 (United States, April 2026)

The Federal Reserve letter states that the revised interagency guidance supersedes and replaces SR 11-7 and SR 21-8. It emphasises a tailored, risk-based approach and is expected to be most relevant to banking organisations with over \$30 billion in total assets. Confirm regulator and entity applicability.

PRA SS1/23 (United Kingdom, 2023)

The PRA supervisory statement sets five principles: model identification and risk classification; governance; model development, implementation and use; independent model validation; and model risk mitigants. Its formal scope is specific, so organisations should confirm applicability while using the principles as a maturity benchmark.

What an interview-ready answer should say

- Do not present every framework as a universal law. State jurisdiction, regulated entity, scope, date, and applicability.
- Know SR 11-7 because enterprise terminology and interview questions still use it, but know that SR 26-2 is the current revised U.S. interagency reference described by the Federal Reserve in 2026.
- Use SS1/23 to explain explicit principles, model tiering, accountable senior management, independent validation, and mitigants.
- Connect supervisory expectations to actual enterprise artifacts: inventory, policy, development document, validation report, approval, monitoring pack, issue log, and committee reporting.

This guide is educational, not legal or regulatory advice. Enterprises must map obligations with their legal, compliance, risk, and supervisory teams.

A practical AI governance framework

Most mature enterprises use a federated model: central policy and standards, business ownership in the first line, independent risk or validation challenge in the second line, and internal audit assurance in the third line. The exact organisation varies; the decision rights should not.

1. Scope and definitions

Define model, AI system, GenAI, agent, high-impact use, third-party AI, end-user computing, and excluded tools. Create an escalation route for uncertain cases.

2. Inventory and ownership

Register model and system components, use cases, versions, owners, users, data, vendors, deployment, geography, dependencies, approvals, and lifecycle status.

3. Risk classification

Rate inherent risk using impact, exposure, purpose, complexity, data, autonomy, scale, vulnerability, legal relevance, reversibility, and detectability.

4. Lifecycle standards

Set risk-tiered requirements for development, procurement, documentation, validation, approval, implementation, monitoring, change, issues, and retirement.

5. Independent challenge

Separate development from validation sufficiently to produce objective challenge. Scale expertise, depth, evidence, and frequency to risk.

6. Control library

Map risks to preventive, detective, and corrective controls with owners, evidence, test method, frequency, thresholds, and escalation.

7. Decision governance

Define who can approve, condition, restrict, reject, accept residual risk, grant exceptions, and require remediation or retirement.

8. Portfolio reporting

Report inventory completeness, risk distribution, validation status, overdue findings, incidents, exceptions, monitoring breaches, vendor exposure, and emerging risks.

Enterprise rule of thumb

A framework is not operational until every requirement has an owner, trigger, evidence artifact, review method, decision right, an escalation path.

Inventory the model and the AI system

Traditional inventories often register one model object. AI governance must also register the end-to-end system and business use. The same foundation model can create very different risk when placed in a low-impact summariser, a customer-facing adviser, or an autonomous agent with write access.

Minimum inventory fields

Identity

System and model name, unique ID, version, lifecycle stage, business unit, legal entity, geography, and environment.

Purpose and use

Decision supported, intended and prohibited uses, users, affected people, deployment channel, materiality, and dependencies.

Ownership

Business owner, model owner, developer, vendor, validator, control owners, approver, monitoring owner, and escalation contact.

Data and technology

Sources, sensitivity, provenance, retention, model or vendor, prompts, retrieval, tools, hosting, and integration points.

Risk and controls

Inherent tier, rationale, regulatory mapping, impact assessment, validations, conditions, limitations, controls, and residual risk.

Operations

Approval date, monitoring metrics and thresholds, incidents, findings, exceptions, changes, review frequency, and retirement plan.

Model versus AI-system scope

Model layer

Algorithm or foundation model, parameters, assumptions, training or adaptation, performance characteristics, and technical limitations.

System layer

Prompts, retrieval, data pipelines, tools, memory, orchestration, access, interfaces, guardrails, monitoring, and vendor dependencies.

Use-case layer

User, decision, business process, affected people, human oversight, incentives, operating environment, legal context, and outcome.

The missing link in many programmes is failure to connect these three layers. A model can be acceptable in isolation while the complete AI system or use remains unsafe, non-compliant, or ineffective.

How to classify model and AI risk

Use a consistent methodology with documented evidence and limited uncontrolled subjectivity. Assess inherent risk before considering controls. Evaluate residual risk after control design and effectiveness. Do not hide a high-impact use by averaging factors into a low score.

Core risk factors

Impact and materiality

Potential financial, customer, safety, legal, prudential, conduct, rights, reputation, or operational consequence.

Purpose and decision

Decision support versus automated decision; advisory versus binding output; intended and reasonably foreseeable misuse.

Exposure and scale

Transaction value, number and vulnerability of affected people, frequency, geography, business criticality, and concentration.

Complexity and uncertainty

Opacity, novelty, assumptions, non-determinism, data limitations, adaptation, interaction effects, and explainability.

Data and rights

Personal or sensitive data, provenance, quality, representativeness, intellectual property, consent, and retention.

Autonomy and access

Ability to select actions, call tools, write to systems, communicate externally, spend money, or change privileges.

Third-party dependence

Vendor transparency, change control, service resilience, subcontractors, audit rights, portability, and exit risk.

Detectability and reversibility

How quickly failure is visible, whether harm can be reversed, strength of human oversight, and recovery options.

Illustrative enterprise tiers

Tiering should change the depth of documentation, independent review, approval authority, monitoring, issue priority, and revalidation. These examples are starting points; each enterprise should calibrate thresholds to its obligations, portfolio, and risk appetite.

Tier 1 - Critical

High-impact or regulated decisions, material financial or rights effects, broad scale, high autonomy, low reversibility, or severe control uncertainty. Requires the deepest independent validation, approval, and monitoring.

Tier 2 - High

Material decision support or customer-facing use with meaningful exposure, complexity, sensitive data, or third-party dependence. Requires formal validation and strong lifecycle controls.

Tier 3 - Moderate

Bounded use, meaningful human review, limited impact, reversible outcomes, and established controls. Apply proportionate documentation, testing, approval, and monitoring.

Tier 4 - Low

Low-stakes assistive use with no consequential decision, limited data, low scale, and easy reversibility. Keep inventory, ownership, acceptable-use controls, and periodic review.

How model documentation is written

A strong development document lets an informed person understand what the model or AI system does, why it was chosen, how it was tested, where it can fail, how it is controlled, and whether it remains fit for purpose. Documentation should describe the implemented system, not an idealised design.

Recommended document structure

- 1 Executive summary**
Purpose, owner, decision, users, risk tier, recommendation, and material limitations.
- 2 Scope and intended use**
Intended, prohibited, and out-of-scope uses; affected groups; operating context; assumptions; and dependencies.
- 3 Architecture and lineage**
Model, data, prompts, retrieval, tools, systems, interfaces, vendors, versions, and end-to-end data flow.
- 4 Data**
Sources, provenance, quality, representativeness, transformations, access, privacy, IP, retention, and known gaps.
- 5 Methodology**
Approach, alternatives, baseline, assumptions, design choices, hyperparameters or configurations, and rationale.
- 6 Testing and results**
Performance, outcomes, robustness, stability, fairness, explainability, safety, security, limitations, and test populations.
- 7 Implementation**
Production verification, thresholds, access, user interface, human oversight, controls, fallback, and reconciliation.
- 8 Monitoring and change**
Metrics, thresholds, frequency, owners, escalation, review triggers, material-change criteria, and rollback.
- 9 Governance record**
Validation, findings, remediation, conditions, approvals, exceptions, residual risk, review date, and sign-off.

What weak documentation looks like

- Generic vendor description with no local use, configuration, data, testing, or limitation analysis.
- High model accuracy but no baseline, threshold rationale, outcome metric, subgroup result, or uncertainty.
- A model card disconnected from production architecture, controls, approvals, monitoring, and change history.
- Known limitations listed without usage restrictions, compensating controls, issue owners, or due dates.

The model validation process

Validation is more than reproducing one metric. It is a risk-proportionate assessment of whether the model and its use are reliable, appropriately implemented, understood, controlled, and suitable for the intended purpose. Independence must support objective challenge, not organisational distance without expertise.

1

Plan and scope

Confirm object, version, intended use, tier, dependencies, previous findings, applicable standards, and validation depth.

2

Review documentation

Assess completeness, internal consistency, traceability, approvals, limitations, and whether the implemented system is described.

3

Assess conceptual soundness

Challenge theory, design, assumptions, alternatives, variables, methodology, thresholds, and the connection to the business decision.

4

Assess data

Challenge lineage, quality, representativeness, leakage, sampling, transformations, privacy, stability, and fitness for use.

5

Verify implementation

Confirm code or configuration, calculations, interfaces, permissions, controls, and production behavior match the approved design.

6

Test performance and outcomes

Reproduce key results, compare baselines and challengers, test errors and slices, assess uncertainty, and connect metrics to outcomes.

7

Test robustness and limitations

Stress unusual conditions, drift, missing data, edge cases, misuse, sensitivity, stability, and failure handling.

8

Assess operations

Review human oversight, monitoring, thresholds, fallback, incident response, change control, user guidance, and retirement.

9

Conclude and report

State scope, evidence, findings, severity, limitations, conditions, compensating controls, residual risk, and a clear validation opinion.

10

Track and revalidate

Monitor remediation and triggers. Revalidate based on risk, time, material change, performance deterioration, incident, or use expansion.

Why AI validation must go further

Traditional MRM remains valuable, but AI can be non-deterministic, adaptive, opaque, vendor-dependent, conversational, multimodal, connected to unstructured data, and capable of taking actions. Validation must therefore expand from model performance to system behavior and business-process outcomes.

Probabilistic outputs

Repeat tests, inspect distributions and worst cases, define acceptable variability, and avoid relying on one successful run.

Foundation-model dependency

Assess inherited risk, vendor evidence, local configuration and use, update behavior, contractual controls, concentration, and exit.

Prompt and context sensitivity

Test instruction hierarchy, prompt variants, long context, multilingual use, user input, and indirect prompt injection.

Retrieval and external data

Validate ingestion, permissions, freshness, chunking, retrieval quality, citations, provenance, poisoning, and deletion.

Agentic behavior

Validate tool selection, arguments, state, permissions, stopping, retries, approvals, recovery, trajectory, and side effects.

Open-ended quality

Use representative tasks, code checks, rubrics, human review, calibrated model graders, disagreement analysis, and outcome metrics.

New harm dimensions

Test bias, toxicity, privacy, security, IP, deception, harmful advice, overreliance, accessibility, and affected-party outcomes.

Continuous change

Version prompts, models, data, retrieval, tools, policies, and vendors; define material-change triggers and regression gates.

A layered validation opinion

- Model layer: Is the model technically understood and suitable for the task?
- System layer: Do data, prompts, retrieval, tools, access, controls, and monitoring work together safely?
- Use-case layer: Is the complete decision process appropriate for the user, affected people, business outcome, and jurisdiction?

Key principle — Do not validate a foundation model once and treat every downstream use as approved. Validate the configured system and intended use, then control expansion and change.

What to test for GenAI, RAG, and agents

GenAI application

- Task success, factual support, completeness, relevance, clarity, refusal, uncertainty, harmful output, privacy, and user reliance.
- Representative scenarios by user, language, complexity, domain, impact, and known failure mode.
- Model, prompt, sampling, guardrail, and fallback comparisons against a documented baseline.

RAG system

- Ingestion completeness, parsing, chunking, metadata, identity-aware filtering, freshness, deletion, and source provenance.
- Retrieval recall, relevance, context precision, answer support, citation correctness, abstention, and unauthorized retrieval.
- Direct and indirect prompt injection, poisoned documents, confidential-data leakage, and stale knowledge.

Agentic system

- End-to-end task success plus trajectory quality: tool selection, argument validity, evidence, unnecessary steps, and recovery.
- Permissions, authentication, authorization, state, memory, idempotency, retries, budgets, stopping, approvals, and side effects.
- Runaway loops, privilege escalation, data exfiltration, unsafe delegation, conflicting agents, tool failure, and partial completion.

Grader design

Code-based

Schemas, exact rules, calculations, tool arguments, permission checks, citations, latency, cost, and deterministic policy requirements.

Model-based

Rubric-scored quality where deterministic checks are insufficient. Calibrate against expert labels and monitor grader bias and drift.

Human

Expert judgment for ambiguous, high-impact, contextual, legal, ethical, or user-experience questions.

Hybrid

Use code for hard constraints, models for scalable rubric scoring, and humans for calibration, exceptions, and high-impact decisions.

Monitoring, change, and issue management

Monitoring pack

Performance and outcomes

Quality, error, calibration, business outcome, overrides, complaints, user feedback, and affected-group results.

Data and drift

Freshness, completeness, schema, provenance, distribution, retrieval corpus, usage, and feature or prompt changes.

Safety and security

Policy violations, prompt injection, leakage, abuse, harmful outputs, access anomalies, and guardrail failures.

Operations

Latency, availability, cost, tool failures, retries, loop length, fallback use, incidents, and vendor service changes.

Control health

Approval status, review dates, evidence freshness, overdue findings, exceptions, training, access review, and reconciliation.

Portfolio concentration

Common vendors, models, datasets, hosting, controls, failure modes, and correlated operational or regulatory exposure.

Material change triggers

- New model, provider, version, fine-tune, prompt strategy, retrieval corpus, tool, memory, data source, threshold, user group, geography, or decision.
- Material performance change, monitoring breach, incident, control failure, new limitation, regulatory change, or expansion beyond approved use.
- Changes should trigger a documented impact assessment and the appropriate level of testing, approval, and revalidation.

Issue management

Every finding should state the risk, evidence, affected scope, severity, required action, accountable owner, due date, interim control, validation of closure, and escalation. Conditions and exceptions should expire; they should not become permanent invisible approvals.

Do not monitor only the model — Monitor the business process and people around it. A stable model can still create harm when usage, incentives, thresholds, populations, data, or downstream actions change.

Where enterprise programmes commonly fail

Inventory without discovery

Teams self-register known systems, but procurement, SaaS, spreadsheets, embedded vendor AI, and employee GenAI remain invisible.

Model-only scope

The model is reviewed, but prompts, RAG, agents, interfaces, human behavior, vendor change, and business outcomes are not.

Principles without controls

Policies say fair, explainable, safe, and accountable, but no owner, evidence, test, threshold, gate, or escalation exists.

Documentation without truth

Templates are complete while actual production configuration, limitations, incidents, dependencies, and workarounds differ.

Validation without outcomes

The validator reproduces technical metrics but does not test the full decision, affected groups, operating process, or real-world benefit.

Vendor assurance without local testing

A provider report is accepted as proof even though the enterprise uses different data, prompts, tools, users, thresholds, and controls.

Tiering without consequences

A risk score is assigned, but it does not change validation depth, approval level, monitoring, issue priority, or review frequency.

Human oversight on paper

A person is named, but lacks time, information, authority, competence, incentives, or a practical way to override.

Change without revalidation

Model, prompt, corpus, tool, or vendor updates enter production without impact assessment or regression evidence.

Committees without ownership

Many stakeholders attend, but no accountable business owner accepts residual risk and no one owns remediation.

The central missing link

The most common gap is translation: turning laws, supervisory principles, and responsible-AI values into lifecycle decisions and testable controls at the model, system, and use-case layers. Mature programmes maintain traceability from obligation or risk to control, owner, evidence, test result, issue, approval, and ongoing monitoring.

A useful challenge question — Show me the exact evidence that this control works for this version, this use, this user population, and this operating environment - and show me who acts when it fails.

What enterprises expect from the roles

Model validator: must know

- MRM policy, scope, model definition, inventory, tiering, lifecycle, validation standards, findings, approval, monitoring, and change triggers.
- Conceptual soundness, data assessment, implementation verification, performance and outcomes analysis, robustness, limitations, and effective challenge.
- AI-specific testing for non-determinism, retrieval, prompt injection, tool use, fairness, explainability, security, human oversight, vendor risk, and monitoring.
- How to write a defensible validation report with evidence, findings, severity, limitations, conditions, residual risk, and a clear opinion.

AI governance professional: must know

- How to translate policy and external obligations into risk-tiered lifecycle requirements, controls, decision rights, evidence, and reporting.
- How to run intake, inventory, classification, impact assessment, vendor review, approval, exceptions, issues, incidents, monitoring, and retirement.
- How first-line owners, developers, validation or risk, legal, privacy, security, compliance, committees, and internal audit work together.
- How to communicate with technical and executive audiences without overstating certainty or treating governance as paperwork.

Depth without coding

A non-coding governance professional should still be able to read an architecture, distinguish a model from a system, understand evaluation design, challenge sampling and metrics, identify leakage and prompt-injection risk, ask how permissions work, interpret monitoring, and judge whether documentation supports the approval. Technical fluency is different from being the production developer.

Strong portfolio artifacts

- AI inventory and risk-tiering methodology with worked examples.
- Model or AI-system development-document template and completed sample.
- Independent validation plan and sample validation report.
- Control library mapped to risks, lifecycle stages, evidence, tests, owners, and escalation.
- Governance dashboard showing risk tiers, validation status, findings, exceptions, incidents, and monitoring breaches.

Model validation interview questions

1. Walk me through an end-to-end model validation.

Strong answer: Start with scope and tier; cover documentation, conceptual soundness, data, implementation, outcomes, robustness, operations, findings, opinion, and revalidation.

2. How do you demonstrate effective challenge?

Strong answer: Show independent evidence, credible alternatives, precise questions, reproducible testing, transparent disagreements, and follow-through on findings.

3. How would you validate a model when outcomes are delayed?

Strong answer: Use leading indicators, stability, sensitivity, proxy outcomes, expert review, backtesting when labels mature, and explicit uncertainty.

4. What is the difference between model performance and model risk?

Strong answer: Performance is one dimension. Risk also depends on purpose, exposure, misuse, implementation, controls, limitations, and consequences.

5. How do you validate a third-party model?

Strong answer: Require sufficient vendor evidence, test local data and use, assess implementation and controls, monitor change, document limitations, and preserve exit options.

6. How would you validate an LLM-based agent?

Strong answer: Test task outcomes and trajectories, tools, arguments, permissions, state, recovery, budgets, approvals, injection, leakage, safety, and monitoring.

7. When is revalidation required?

Strong answer: Use risk-based frequency plus triggers such as material change, performance breach, incident, new use, new population, vendor update, or control failure.

8. What makes a validation finding strong?

Strong answer: Clear risk and evidence, affected scope, severity rationale, required outcome, owner, due date, interim control, and closure test.

AI governance interview questions

1. How would you build an enterprise AI governance framework?

Strong answer: Define scope, inventory, tiering, lifecycle, controls, roles, validation, approvals, monitoring, issues, reporting, training, and continuous improvement.

2. How do you classify AI risk?

Strong answer: Assess inherent impact, decision, exposure, scale, data, complexity, autonomy, third parties, legal relevance, detectability, and reversibility; then assess residual risk.

3. How do SR 11-7, SR 26-2, and SS1/23 relate?

Strong answer: State jurisdictions and dates. Explain SR 11-7's legacy vocabulary, SR 26-2's 2026 replacement and risk-based tailoring, and SS1/23's five U.K. principles.

4. What is the difference between MRM and AI governance?

Strong answer: MRM focuses on model risk. AI governance is broader and includes full systems, use cases, rights, safety, security, privacy, vendors, accountability, and regulation.

5. How do you govern a foundation model you cannot inspect fully?

Strong answer: Assess provider and local evidence, constrain use, test the configured system, monitor changes, define contractual controls, document uncertainty, and maintain exit and incident paths.

6. How do you make human oversight meaningful?

Strong answer: Give the reviewer timely information, competence, authority, time, clear thresholds, an override path, feedback, and monitoring of real outcomes.

7. What metrics show an AI governance programme works?

Strong answer: Inventory coverage, risk-tier quality, review time, control effectiveness, validation status, overdue issues, incidents, exceptions, outcomes, audit findings, and safe adoption.

8. A business team wants to launch before validation is complete. What do you do?

Strong answer: Assess risk and missing evidence, define whether use can be limited safely, require compensating controls and explicit approval, or stop launch when residual risk is unacceptable.

9. What is the biggest gap in enterprise AI governance?

Strong answer: Often it is traceability from risk or obligation to working control, evidence, test, owner, decision, and monitoring across the model, system, and use.

10. How would you explain AI governance to executives?

Strong answer: Frame business value, material exposure, decision rights, control coverage, residual risk, portfolio concentration, incidents, and the decisions leadership must make.

Before an AI system is approved

Check 1

The system and use are identified in inventory with current versions, dependencies, geography, users, and accountable owners.

Check 2

The intended decision, benefit, affected people, prohibited uses, and operating context are clear.

Check 3

Inherent risk and applicable obligations are assessed with a documented tier rationale.

Check 4

Development or procurement documentation describes the implemented model, system, data, configuration, testing, limitations, and controls.

Check 5

Independent validation or review is complete at the depth required by risk, with findings and limitations recorded.

Check 6

Data, privacy, security, fairness, safety, legal, compliance, vendor, and operational reviews are complete where applicable.

Check 7

Human oversight is practical and tested, not only named in policy.

Check 8

Monitoring metrics, thresholds, owners, escalation, incident response, fallback, and change triggers are approved.

Check 9

Conditions, exceptions, residual risk, compensating controls, and decision authority are explicit.

Check 10

Users receive clear instructions, limitations, escalation paths, and role-appropriate AI literacy.

The approval test

Can the accountable owner explain what the system does, who can be affected, what can go wrong, why the evidence is sufficient, which controls are working, what remains uncertain, and who will act if performance or risk changes?

Research basis and source notes

This guide is an original educational synthesis. It does not replace legal advice, supervisory interpretation, an organisation's policy, or role-specific technical procedures. Confirm current applicability, scope, and implementation with qualified legal, compliance, risk, and supervisory teams.

Federal Reserve: SR 26-2, Revised Guidance on Model Risk Management (2026)

<https://www.federalreserve.gov/supervisionreg/srletters/SR2602.htm>

Current Federal Reserve supervisory letter describing revised interagency MRM guidance and its replacement of SR 11-7 and SR 21-8.

Federal Reserve: Revised Supervisory Guidance on Model Risk Management

<https://www.federalreserve.gov/frs/guidance/supervisory-guidance-on-model-risk-management.htm>

Risk-based MRM, development and use, validation and monitoring, governance, inventory, documentation, and third-party products.

Federal Reserve and OCC: SR 11-7 attachment (2011)

<https://www.federalreserve.gov/boarddocs/srletters/2011/sr1107a1.pdf>

Historical foundational guidance on development, implementation and use, validation, and governance. Superseded by SR 26-2 as stated by the Federal Reserve.

PRA: SS1/23 Model risk management principles for banks

<https://www.bankofengland.co.uk/-/media/boe/files/prudential-regulation/supervisory-statement/2023/ss123.pdf>

Five principles covering identification and classification, governance, development and use, independent validation, and mitigants.

NIST AI Risk Management Framework

<https://www.nist.gov/itl/ai-risk-management-framework>

Voluntary framework structured around Govern, Map, Measure, and Manage for trustworthy and responsible AI risk management.

NIST AI RMF: Generative AI Profile

<https://www.nist.gov/itl/ai-risk-management-framework/ai-risk-management-framework-generative-artificial-intelligence-profile>

Generative-AI-specific risks and actions that extend general AI risk management.

European Commission: AI Act

<https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai>

Official overview of the EU risk-based AI regulatory framework, roles, prohibited practices, and high-risk and transparency requirements.

ISO/IEC 42001: Artificial intelligence management systems

<https://www.iso.org/standard/81230.html>

Management-system standard for establishing, implementing, maintaining, and continually improving responsible AI governance.

Final reminder

Strong AI governance connects policy to decisions, controls, evidence, and outcomes. Strong validation connects model behavior to system reliability, intended use, affected people, and residual risk.