fairplay

FairPlay's Model Risk Management Manual for Agentic Systems:

A Roadmap to Regulatory Readiness

Introduction

This Manual is intended to help developers and users of agentic systems—including model governance teams, compliance officers, and other financial institution personnel—strengthen their MRM practices by addressing the unique operational, compliance, and reputational risks these systems present.

The guidance in this manual builds on the Federal Reserve's SR 11-7 supervisory letter on Model Risk Management, while integrating emerging best practices in agentic governance, generative AI oversight, and financial services compliance. It incorporates lessons from practical AI governance frameworks, industry initiatives on responsible AI, and large-scale red-teaming efforts that expose common vulnerabilities in deployed generative systems. Although SR 11-7's core principles—governance, conceptual soundness, and ongoing monitoring—remain foundational, their application to generative and agentic systems requires reinterpretation given their dynamic, multi-step, and tool-enabled nature. This manual is not a substitute for established MRM practices; it is designed to augment them with tools to address the novel risks agents introduce.

This Manual provides:

- **Key questions** that Model Risk Management teams should ask as part of evaluating an AI agent deployed by a bank, fintech, third-party service provider, or other regulated institution.
- **Context and risk framing** for each examination area, explaining why these questions matter and what underlying risks they aim to address.
- **Specific testing approaches** that institutions can use to validate that their agentic systems are fit for purpose in a given financial application, including methods for assessing factual accuracy, hallucination risk, data leakage, prompt quality, fairness, and resilience under stress.

By following these steps, financial institutions can validate AI systems effectively, strengthen internal controls, build confidence in deployments, mitigate reputational risk, prevent consumer harm, and demonstrate to stakeholders—including regulators—that their systems operate safely and as intended. Although agentic governance remains an emerging field, these incremental measures can bring order to uncertainty. At the same time, the Manual stresses that validation is necessary but not sufficient: sustaining trust in agentic systems over time requires continuous monitoring, robust fallback planning, and strong independent challenge functions.

1 Governance, Policies, and Procedures

Context & Background

Strong governance is the first line of defense in SR 11-7. For agentic systems, governance must address the **full lifecycle of oversight**—from initial design and approval, through deployment, monitoring, and eventual decommissioning—because the risks they present can evolve over time.
Agentic systems, particularly those with **tool-use capabilities** (e.g., the ability to query databases, trigger transactions, or update records), expand the scope of potential harm if not carefully controlled. Effective governance requires defining parameters such as scope of action (what the agent is and is not allowed to do), escalation authority (who can intervene and under what conditions), and policy enforcement (technical and procedural safeguards to keep the agent within approved boundaries).

Governance also requires **technical controls** such as:

- **Tool access permissions:** Configuring "least privilege" so that an agent only has access to the specific tools necessary for its tasks.
- **Audit logging of tool use:** Recording every tool invocation with time stamps, parameters, and outcomes, so that any action can be traced and reviewed later.

Many financial institutions rely on **"human-in-the-loop"** oversight—keeping a human reviewer in control of approving AI outputs or actions. However, this approach faces well-documented limitations:

- **Vigilance fatigue –** Over time, humans monitoring a mostly well-behaved system tend to experience declines in attention and accuracy, missing subtle or rare problems. When events requiring intervention are rare, human operators become less likely to detect them when they do occur, often because their sensitivity to infrequent events diminishes over time.
- **Automation bias –** The tendency to over-trust and accept automated outputs as correct, even in the absence of verification, particularly when the system has historically performed well.
- **Manual delay -** waiting for human-in-the-loop reviews slows down the speed of delivery, which is arguably one of the main reasons for using AI agents.

These three factors can erode human oversight, allowing flawed or risky outputs to pass unchecked. While some aspects of agentic performance are inherently subjective (e.g., the quality of an explanation), governance frameworks should explicitly account for them. Wherever possible, they should be complemented with objective thresholds—such as error rates for binary responses or latency limits—that, when breached, trigger predefined and automated escalation actions.

Finally, governance should require a continuity plan with a tested fallback strategy—such as maintaining an alternative model or process that can replace the primary agent if performance falls below acceptable levels. This strategy should be operationalized and tested in advance, not left as a hypothetical, to ensure a seamless takeover without disruption.

Key MRM Questions

- Does the institution's model governance framework explicitly address lifecycle risks unique to GenAI, including vigilance fatigue, automation bias, and the delays caused by manual oversight?
- Does it include governance for **tool access permissions** and **audit logging** of all agent tool use?
- Are escalation protocols tied to **objective performance metrics** with predefined cutoffs for disabling the system?
- Is a **continuity plan or fallback strategy** documented, maintained, and tested on a reasonable cadence?
- Has the board and senior management received **specialized GenAI training** on operational, compliance, and ethical risks?

Sample Tests

Governance Drill Simulation

- **What:** Tests the institution's ability to detect a governance breach (e.g., policy violation, risk threshold exceeded) and escalate it according to protocol.
- **How:** Simulate a scenario in which the agent produces a non-compliant output or exceeds a risk threshold; observe the effectiveness of controls, including escalation actions, decision-making, fallback activation, and post-incident reporting
- **Evidence:** Logs of the simulated incident, escalation chain records, decision rationale, and post-mortem review notes.

Tool-Use Access & Logging Review

- **What:** Ensures that the agent has the minimum necessary tool access and that all tool use is logged for auditability.
- **How:** Review tool permission settings, applicable controls, conduct a test of agent tool invocation, and inspect generated logs.
- **Evidence:** Tool permission matrices, sample log excerpts, change history of tool permissions, and log retention policy.

Board Training Audit

- **What:** Verifies that the board and senior management have received training on GenAI-specific risks, including the cognitive factors above.
- **How:** Review attendance records, agendas, and training curricula.
- **Evidence:** Signed training completion certificates, agendas, attendance records, and training materials.

2 Model Definition and Inventory

Context & Background

Under SR 11-7, a financial institution must maintain a complete and accurate inventory of all models in use. This inventory allows internal and external stakeholders to understand the scope of model risk, ensure proper governance, and manage dependencies.
For agentic systems, “model” must be defined broadly. It is not only the foundation model (e.g., GPT-4, Claude) but also:

- Any fine-tuning datasets applied to specialize the model for institutional use; Policy overlays;
- Retrieval-Augmented Generation (RAG) components, including the corpora from which information is drawn;
- Prompt templates or prompt libraries that form part of the model’s logic;
- Tool integrations the agent can invoke (databases, APIs, calculators, transaction systems); and
- Plugins or middleware that transform inputs or outputs.

A frequent problem—highlighted in both regulator commentary and red-teaming research—is that many AI-enabled processes are not recognized internally as “models” because they operate within business units without formal MRM oversight. This leads to “shadow AI”—systems making important decisions outside governance controls.

Maintaining an up-to-date inventory is more than a compliance exercise:

- It supports version control, so outputs can be traced to the exact model state used at the time;
- It allows concentration risk analysis (e.g., reliance on a single vendor); and
- It helps identify integration points where changes could have unintended consequences.

Without this inventory discipline, regulators and other stakeholders may conclude that the institution does not have effective model risk governance—regardless of how strong individual controls are elsewhere.

Key MRM Questions

- Does the institution's model inventory clearly identify each agentic system in use, including:

  - The foundation model(s) powering the system;
  - Overlays;
  - Any fine-tuning datasets or methods used;
  - RAG retrieval components and their underlying data sources;
  - Prompt templates or prompt libraries; and
  - All tools, APIs, and integrations the agent can invoke.

- Is the inventory updated in **real time** (or near-real time) when any of these components change?

- Are Agents **classified by criticality and risk level**, with higher-risk systems subject to more rigorous controls?

- Does the institution's definition of "model" explicitly include AI-enabled processes that may otherwise escape oversight (avoiding "shadow AI")?

Sample Tests

Inventory Completeness Check

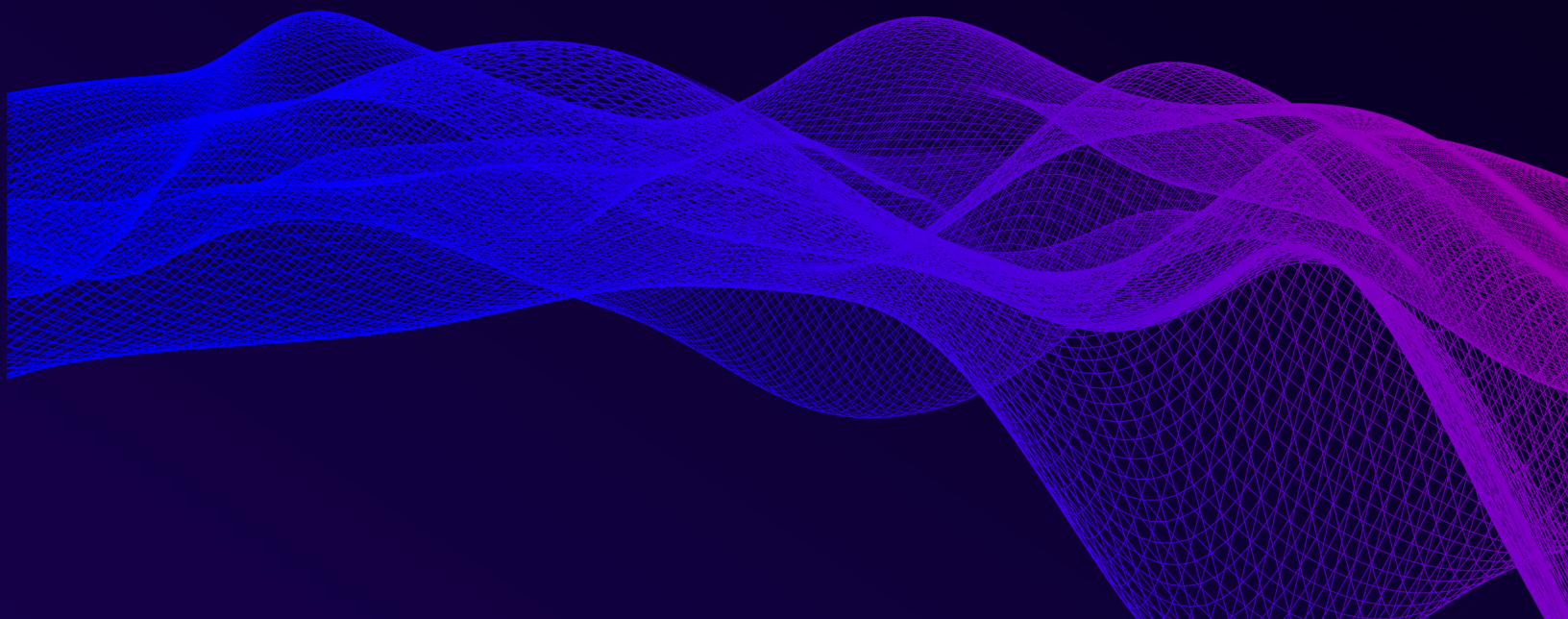
- **What:** Confirms that all agentic systems and their components are documented in the model inventory.
- **How:** Compare system access logs, vendor contracts, and business unit project lists against the official model inventory to identify any unlisted systems ("shadow AI").
- **Evidence:** Annotated comparison report showing match/mismatch, updated inventory list, remediation plan for any omissions.

Dependency Mapping

- **What:** Maps all dependencies (feeder models, datasets, etc.) for each agentic system to reveal concentration risk and change impact points.
- **How:** For each inventoried model, create a diagram showing foundation model vendor, fine-tuning datasets, RAG corpora, tools, APIs, and downstream systems affected by outputs.
- **Evidence:** Dependency diagrams, vendor dependency register, and change impact analysis reports.

Inventory Change Audit

- **What:** Verifies that changes to model components are logged and trigger updates to the inventory.
- **How:** Review the last 6 months of model updates (e.g., new prompts, tool access changes, RAG corpus refreshes) and confirm that each was reflected in the official inventory.
- **Evidence:** Change tickets, dated inventory entries, and approval logs for each update.



3 Conceptual Soundness

Context & Background

Under SR 11-7, conceptual soundness refers to how well the design of a model aligns with its intended purpose, the appropriateness of its assumptions, and the validity of its methodology. For traditional statistical models, this often means assessing the choice of variables, mathematical form, and underlying theory.

For agentic systems, conceptual soundness takes on a new dimension because:

- gents are semi-autonomous and non-deterministic. Their design must anticipate variability in behavior, providing enough foresight and flexibility for the agent to make decisions within defined boundaries while still accommodating a range of possible outcomes and unexpected behaviors—much like human-centered systems account for judgment, error, and adaptation.
- "Design" encompasses concrete architectural choices. These include whether the system relies solely on a large language model or incorporates a retrieval-augmented generation (RAG) framework—decisions that shape how the agent processes information, reasons about queries, and balances accuracy with controllability.
- Prompting strategy drives behavior. Instructions, role definitions, and guardrails directly influence the agent's outputs.
- Retrieval logic shapes performance. In RAG systems, the retrieval mechanism determines which sources are consulted and how they are ranked.
- Boundaries define the safe operating envelope. Constraints on tool use, data access, and action-taking establish the limits within which the agent can safely function.

Failure to establish conceptual soundness can lead to:

- Hallucinations – when an AI system generates plausible-sounding but factually incorrect, unverifiable, or fabricated information, such as fake citations, wrong numbers, or confident statements about non-existent events.
- Bias – where model behavior systematically disadvantages certain groups or yields inconsistent decisions.
- Off-policy behavior – where the agent takes actions outside its intended scope, possibly creating compliance violations, or even costing the company money directly (i.e., waiving fees the agent was not supposed to waive or reimbursing customers in excess of what they paid).

All of which can lead to reputational and financial risk, in addition to regulatory risk.

RAG is particularly relevant here. In RAG, the model retrieves relevant documents or facts from a curated knowledge base to inform the LLM's thinking before generating its answer. This can reduce hallucinations, but only if the retrieval logic is sound and the sources are trustworthy.

Because small context changes can lead to very different LLM outputs, conceptual soundness for agentic systems is not just a pre-deployment step—it requires anticipating how the system will behave in varied, sometimes adversarial, scenarios.

Key MRM Questions

- Has the institution documented **why** it chose a particular architecture (e.g., general LLM vs. LLM+RAG, and the enabling tools or infrastructure) for the given use case, including trade-offs in accuracy, transparency, and control?
- Are the **known limitations of the foundation model**—including newly identified issues such as publicized accuracy concerns for certain functions, domain gaps, or exposed biases—being tracked, documented, and communicated to governance teams?
- Have prompt templates, retrieval strategies, and guardrails been designed to increase accuracy, **minimize hallucinations** and reduce bias?
- Are operational boundaries for the agent clearly defined (e.g., which actions it may take autonomously vs. which require human approval)?
    - Are those boundaries tested? If so, how?
- Has the institution tested the agent's behavior under **adverse conditions**—including ambiguous queries, incomplete data, and contradictory or malicious inputs?

Sample Tests

Scenario Simulation

- **What:** Assesses whether the agent behaves consistently and appropriately under a range of realistic and edge-case scenarios.
- **How:** Present the agent with scripted scenarios, including well-defined queries, ambiguous requests, adversarial prompts, and misleading context. Evaluate accuracy, compliance, and adherence to scope boundaries.
- **Evidence:** Scenario scripts, annotated results, pass/fail analysis for each scenario, and documented design changes made in response.

Comparative Baseline Testing

- **What:** Compares the performance of the chosen architecture against one or more alternative designs (e.g., LLM-only vs. LLM+RAG).
- **How:** Run identical test sets on each architecture and compare outputs for accuracy, groundedness, and compliance.
- **Evidence:** Side-by-side test results, performance metric summaries, and rationale for final architecture choice.

Hallucination Stress Test

- **What:** Measures the agent's propensity to produce ungrounded or fabricated outputs.
- **How:** Provide prompts that invite speculation, include incomplete data, or request citations. Verify factual grounding against known truth sets or retrieved sources.
- **Evidence:** Hallucination rate statistics, examples of problematic outputs, and mitigation steps taken.

4 Data and Training Transparency

Context & Background

In SR 11-7, regulators expect financial institutions to understand what data a model was trained on and to ensure that data is appropriate, accurate, and legally permissible for its intended use.
With foundation models, this expectation becomes challenging because the institution usually does not control or have complete visibility into the data used for initial training. For example, a large language model might have been trained on a mixture of public web data, licensed content, and other proprietary sources—details of which are often only partially disclosed by the vendor.

However, the institution can and must exercise full transparency and control over:

- Fine-tuning datasets – Any data added to adapt the model to the institution's domain (e.g., customer service transcripts, underwriting rules).
- RAG retrieval sources – The documents or databases the agent queries in real time to produce answers.
- Prompt libraries – If prompts include sensitive or proprietary information, their origin and approval should be tracked.
- Any "source of truth" data sources used in the development and testing of the agent.

Data transparency matters for multiple reasons:

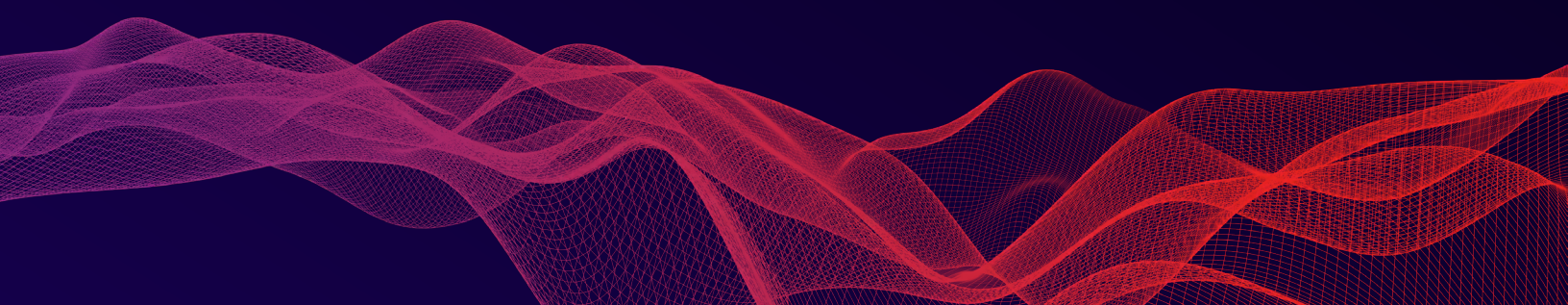
- Data fit - The data used to design, test, and operate an agent must align with its intended purpose. Data that is accurate in one context may be misleading in another. For example, "source of truth" AML data from a regional credit union might not adequately support an agent designed for global banking use cases.
- Regulatory compliance – The use of certain data types (e.g., consumer credit data, personally identifiable information) is governed by strict laws.
- Reputation and intellectual property risk – Using copyrighted or confidential data without rights can trigger legal or reputational fallout.
- Bias mitigation – Without visibility into the composition of fine-tuning or retrieval data, it may be challenging for institutions to assess whether the data reflects the diversity of real-world users and use cases, or whether it embeds hidden biases.

Key concepts explained:

- Data provenance – The complete record of where data came from, how it was collected, and any transformations applied.
- Data rights – Legal authority to use the data for training, fine-tuning, or retrieval.
- Knowledge source quality metrics – Measures such as accuracy, completeness, and timeliness for RAG sources.

Key MRM Questions

- Has the institution documented all **fine-tuning data sources**, including their provenance, rights, and quality checks?
- Has the vendor provided any disclosures or attestations about the **foundation model's** training data, and has the institution assessed these for potential compliance or reputational risks?
    - Could the data be inaccurate or biased?
    - Are the data representative? Do the datasets used for training, fine-tuning, or retrieval adequately reflect the diversity of populations, scenarios, and contexts in which the agent will be deployed? For example, do they cover relevant demographic groups, geographic regions, and edge cases, or do gaps in coverage risk producing biased or unreliable outcomes?
- For RAG systems, are all **retrieval sources** cataloged, with quality metrics and periodic reviews to ensure accuracy and freshness?
- Is there a documented control and/or process to vet new or updated data sources before they are incorporated into the agent's retrieval or fine-tuning workflow?
- Does the institution have controls and/or policies preventing the ingestion of sensitive, confidential, or legally restricted data into fine-tuning or RAG systems without explicit approval?



Sample Tests

Fine-Tuning Data Audit

- **What:** Verifies that all fine-tuning datasets are documented, rights-cleared, and quality-checked.
- **How:** Review the data source inventory, confirm legal permissions, and check quality metrics such as accuracy rates, label consistency, and representativeness.
- **Evidence:** Data source list with provenance details, rights documentation (licenses, contracts, internal approvals), and data quality reports.

Vendor Attestation Review

- **What:** Assesses disclosures from the foundation model vendor regarding pre-training data.
- **How:** Compare vendor-provided information against internal compliance checklists; flag potential high-risk categories (e.g., unlicensed copyrighted material, regulated data).
- **Evidence:** Vendor attestation documents, internal review notes, and any follow-up correspondence with the vendor. *Remember, that **a vendor attestation does not shield a financial institution from liability**; the institution shares the risk if the vendor violates its legal or regulatory obligations to the customer.

Knowledge Source Quality Assessment (for RAG)

- **What:** Confirms that all retrieval sources meet institutional accuracy and reliability standards.
- **How:** Sample retrieved documents for correctness, verify they are current, and measure retrieval precision/recall against a known truth set.
- **Evidence:** Quality metric reports, sampling logs, and documented remediation for any low-quality sources.

Data Leakage Prevention Check

- **What:** Ensures that sensitive or restricted data is not inadvertently included in fine-tuning or retrieval sources.
- **How:** Use automated scanning tools to detect PII, PCI, PHI, or other restricted content in data repositories; review against approved use lists.
- **Evidence:** Scan reports, exception lists, and remediation actions.

5 Model Validation and Independent Review

Context & Background

In SR 11-7, model validation is the rigorous process of determining whether a model is appropriate for its intended use before it is put into production. This includes evaluating conceptual soundness, testing performance under realistic conditions, and assessing limitations or weaknesses.
For agentic systems, validation must go beyond traditional accuracy checks to include behavioral, security, and resilience testing. Unlike static statistical models, LLM-based agents can generate different outputs for the same input depending on context, recent interactions, or updated parameters.

Key challenges for validation in agentic systems include:

- Adversarial prompt testing – Probing the agent with inputs deliberately designed to bypass guardrails or cause policy violations.
- Outlier interaction testing – Seeing how the system responds to edge cases, incomplete data, or unusual phrasing.
- Data leakage testing – Ensuring the agent does not disclose sensitive or proprietary information when prompted in creative ways.
- Drift detection – Identifying changes in performance over time due to vendor model updates, fine-tuning adjustments, or evolving data.

An independent review—performed by a group separate from the development team—is critical because:

- Vendor guardrails may not align with the institution's compliance obligations.
- Developers may be unconsciously biased toward interpreting test results as favorable.
- Independent parties may have a broader view of the kinds of risks that tend to arise with agents at other institutions, and are able to bring that awareness to the review of the agent as part of the validation process.

Ensure that any reviewers have the necessary technical sophistication to understand and pressure-test your vendors.

Key MRM Questions

- Has the pre-deployment validation process included **both normal-use and stress-testing scenarios**, including adversarial prompts and outlier cases?
- Has the institution tested for **data leakage risks**, such as the ability to extract sensitive internal information from the agent?
- Are validation results **repeatable**—do the same test cases produce consistent performance over time?
- Was the validation conducted by an **independent team** with no stake in the deployment decision?
- Are **monitoring metrics** and controls established during validation so that post-deployment monitoring can track for drift or degradation?

Sample Tests

Hallucination Testing

- **What:** Detects inconsistent answers, which can indicate hallucination.
- **How:** The LLM is prompted multiple times with the same (or slightly varied) questions. Factual claims should remain consistent across generations, while hallucinated content will likely vary or be inconsistent. The testing can include misleading questions or false premises designed to provoke hallucination (e.g. asking to explain fictional regulations).
- **Evidence:** Test dataset; semantic similarity metric, like BERTScore, to compare sentences within the different responses; annotated examples of incorrect outputs, and documentation of mitigation steps.

Data Leakage Testing

- **What:** Detects whether the agent can be tricked into revealing confidential or sensitive data.
- **How:** Use “prompt injection” techniques—malicious or cleverly worded prompts—to attempt extraction of internal information. Also test for context-bleed, where the agent inadvertently recalls sensitive details from prior interactions.
- **Evidence:** Injection prompt library, test transcripts, and incident reports for any successful leakage.

Prompt Robustness Testing

- **What:** Assesses how well the agent maintains accuracy and compliance when prompts are rephrased, contain typos, or include misleading context.
- **How:** Create variants of test prompts, including noisy or adversarial inputs, and measure performance variance.
- **Evidence:** Prompt variation set, output accuracy comparisons, and documented adjustments to prompt templates or retrieval logic.

Independent Validation Review

- **What:** Confirms that validation activities are performed and reviewed by staff independent from the development team, and with the necessary technological skills to critically examine and validate.
- **How:** Review organizational charts, credentials, interview validation and development teams, and confirm separation of roles.
- **Evidence:** Validation reports signed by independent reviewers, meeting notes from validation sign-off, and governance approvals.

6 Ongoing Monitoring

Context & Background

Under SR 11-7, ongoing monitoring is the continuous process of assessing a model's performance, stability, and compliance once it is in production. This ensures that models continue to function as intended and remain within risk tolerances over time.

For agentic systems, ongoing monitoring is more critical—and more complex—than for traditional models because:

- Outputs can change without any code modifications when the foundation model is updated by the vendor.
- User interactions can alter short-term model behavior (e.g., through conversation history or context memory).
- Emerging risks, such as hallucinations, bias, and data leakage, can appear after deployment, even if pre-deployment validation was thorough.

One major challenge is that many institutions rely heavily on human-in-the-loop oversight to catch issues. While human review is essential, it is prone to the limitations discussed above.

To mitigate these risks, monitoring of agentic systems should combine human-led oversight with AI-assisted "check-the-checker" mechanisms—for example, pairing human review with an auditing agent that validates automated checks and continuously scans outputs against:

- Assertion-based compliance rules – Predefined "must be true" statements that outputs can be tested against (e.g., "The agent must not provide investment advice without a disclosure statement").
- Factual accuracy checks – Verifying key factual elements against trusted internal or external databases.

Pairing human oversight with independent AI-assisted oversight, and comparing their respective findings for consistency can help reduce false-negatives and augment the accuracy and volume-coverage of the human-in-the-loop approach.

Key MRM Questions

- Does the monitoring process combine **human review** with **automated, AI-assisted checks** to reduce vigilance decrement and automation bias?
- Are compliance checks **objective** (measuring outputs against objective rules) rather than open-ended human judgment alone?
- Are **factual accuracy checks** run continuously or on a frequent, scheduled basis, and are results tracked over time?
- Are these controls documented, tested, and updated on a regular basis?
- Are monitoring results fed into **dashboards** with clear escalation triggers for intervention?
- Is there a documented process for updating monitoring rules when regulatory requirements, product features, or risk profiles change?

Sample Tests

Assertion-Based Compliance Scanning

- **What:** Confirms that automated monitoring is checking outputs against a set of objective compliance rules.
- **How:** Define compliance assertions, run them on a representative sample of outputs, and review pass/fail rates.
- **Evidence:** Assertion list, scan results, flagged violations, and documented remediation steps.

Factual Accuracy Monitor

- **What:** Continuously verifies the correctness of factual outputs against a trusted and objective source.
- **How:** Select high-priority facts (e.g., product rates, policy terms) and compare agent outputs to a source-of-truth database.
- **Evidence:** Accuracy reports, historical accuracy trend graphs, and remediation logs.

Latency & Load Monitoring

- **What:** Tracks whether the system maintains performance under varying loads, which can impact accuracy and reliability.
- **How:** Test response times and output quality under both normal and peak loads. Define clear quantitative performance benchmarks, document controls, and update them through regular load and stress testing.
- **Evidence:** Latency trend reports, load test results, and any adjustments made to maintain service levels.

Update Regression Testing

- **What:** Detects unintended changes in behavior after vendor model updates or configuration changes.
- **How:** Re-run a standard set of test prompts before and after updates; compare results for accuracy and compliance.
- **Evidence:** Pre- and post-update test logs, change impact assessments, and documented mitigations.

7 Outcome Analysis and Fairness Testing

Context & Background

Under SR 11-7, outcome analysis is the process of evaluating a model's real-world performance—not just whether it is accurate, but whether its outputs lead to fair and compliant outcomes in practice.

For agentic systems in financial services, this means looking beyond technical accuracy to ensure that decisions, recommendations, or customer interactions do not produce unintended disparities across legally protected groups or other sensitive categories.

Key fairness-related risks include:

- Disparate treatment – When similarly situated individuals are treated differently based on protected class status.
- Disparate impact – When a policy or system that is neutral on its face disproportionately harms members of a protected class (race, sex, age, etc.), even without intentional discrimination.
- Groundedness – Whether the agent's answers are based on verifiable sources, especially for RAG systems. Outputs that are ungrounded or speculative can create compliance and reputational risks.
- Consistency – Ensuring that similar queries produce similar, fair results, regardless of who is asking or how the question is phrased.

Because agentic systems can update dynamically and are highly context-sensitive, outcome testing should be ongoing rather than one-time. This is especially true for RAG architectures, where the source data may change, or for systems whose outputs vary due to prompt structure, user profile, or session history.

Key MRM Questions

- Are regular **fairness analyses** performed on the agent's outputs to detect biases across protected classes (e.g., race, gender, age) or close proxies for protected classes (i.e., geography)?
- Are **consistency checks** conducted to ensure similar queries yield similar outputs regardless of who is asking or what demographic information is provided?
- For **RAG systems**, are outputs routinely tested for **groundedness**—that is, whether they can be traced to and supported by authoritative sources?
- Are **benchmark tests** used to measure whether the current production system continues to meet fairness and performance expectations?
- Is there a documented escalation and remediation process when fairness or consistency issues are detected?

Sample Tests

Demographic Parity Simulation

- **What:** Evaluates whether outputs or decisions differ inappropriately across demographic groups.
- **How:** Provide simulated profiles that differ only by protected class attributes. Generate outputs, translate them into structured outcomes (e.g., decision type, recommendation strength, sentiment), and compare across groups.
- **Evidence:** Defined test cases and profiles, documentation of how unstructured outputs were mapped into measurable categories (e.g., sentiment, recommendation type, decision proxy), statistical results (e.g., adverse impact ratio), and remediation steps for any disparities.

Consistency Testing Across Variants

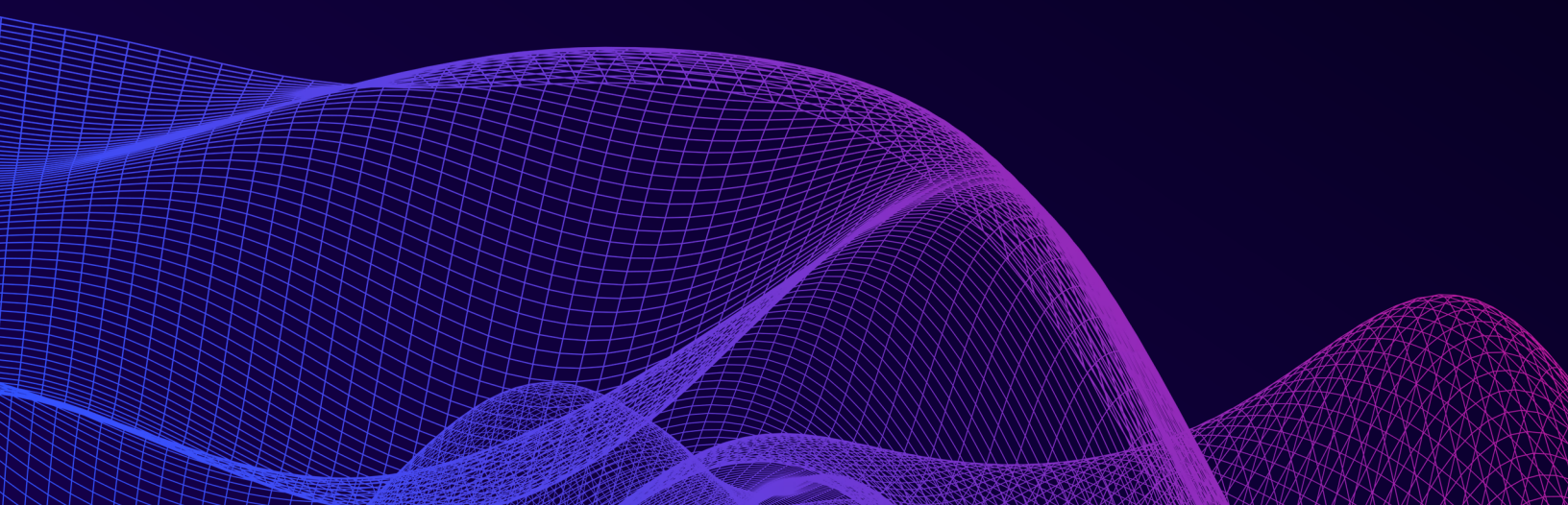
- **What:** Ensures the agent's behavior is stable across demographic variants and query phrasings.
- **How:** Pose equivalent questions with slight variations in wording, demographics, or context; measure changes in output quality and fairness metrics.
- **Evidence:** Variant prompt library, output comparison tables, and corrective action logs.

RAG Groundedness Test

- **What:** Confirms that responses from retrieval-augmented generation systems are supported by their retrieved sources.
- **How:** Sample responses, extract cited sources, and verify that the source material supports the claims made by the agent.
- **Evidence:** Annotated output samples, groundedness scoring sheets, and retraining or corpus-cleaning records where failures occurred.

Champion–Challenger Evaluation

- **What:** Compares the production agent to an alternative model or configuration to detect performance or fairness drift.
- **How:** Run parallel tests using the same prompt set on both models; compare fairness and accuracy metrics.
- **Evidence:** Side-by-side comparison reports, decision logs, and governance approvals for model changes.



8 Third-Party Risk Management

Context & Background

In SR 11-7 and related OCC guidance, third-party risk management means that a financial institution is responsible for ensuring that vendors, service providers, and other third parties meet the same risk, compliance, and operational standards the institution would apply to its own operations. Federal financial regulators have also issued guidance that financial institutions are responsible for the actions of their vendors; while work may be outsourced, liability cannot be.

For agentic systems, this is especially critical because:

- Many financial institutions rely on external vendors for the foundation models or hosted services that power their agents.
- Vendors may update the underlying model or guardrails without notice, potentially altering the agent's behavior.
- The institution may have limited visibility into the model's training data, architecture, or internal safety features.

Two key concepts here are:

- Vendor guardrails – Safeguards built into a third-party system to prevent unsafe, non-compliant, or inaccurate outputs. These may include content filters, safety classifiers, or usage restrictions. Institutions cannot assume these are sufficient or aligned with their own regulatory obligations.
- SLAs (Service Level Agreements) – Contracts that define the performance, availability, incident reporting, compliance, and documentation requirements the vendor must meet. For GenAI systems, SLAs should also cover model updates, explainability documentation, and prompt/response logging.

Because regulators hold the financial institution—in addition to the vendor—responsible for model performance and compliance, independent verification of vendor claims and controls is essential.

Key MRM Questions

- Has the institution formally assessed each GenAI vendor's **alignment with SR 11-7 principles**, including governance, validation, and monitoring practices.
- Is the vendor aware of its regulatory obligations?
- Are **Service Level Agreements** in place that require vendors to:
  - Provide advance notice of model updates or changes,
  - Disclose relevant documentation, and
  - Report incidents within defined timelines?
- Has the institution conducted oversight and **independent testing** of vendor guardrails to verify they are effective in preventing non-compliant outputs?
- Is there a process for **regular re-assessment** of vendor performance, security posture, and compliance alignment?
- Are there **exit plans** in place if a vendor no longer meets contractual or compliance requirements?

Sample Tests

Guardrail Penetration Test

- **What:** Evaluates whether vendor-implemented safety features can be bypassed.
- **How:** Attempt to trigger prohibited outputs using adversarial prompts or obfuscation techniques; test known failure modes such as prompt injection, context poisoning, or indirect queries.
- **Evidence:** Test scripts, transcripts of attempted bypasses, incident logs, and documentation of corrective actions taken with the vendor.

Contract Compliance Check

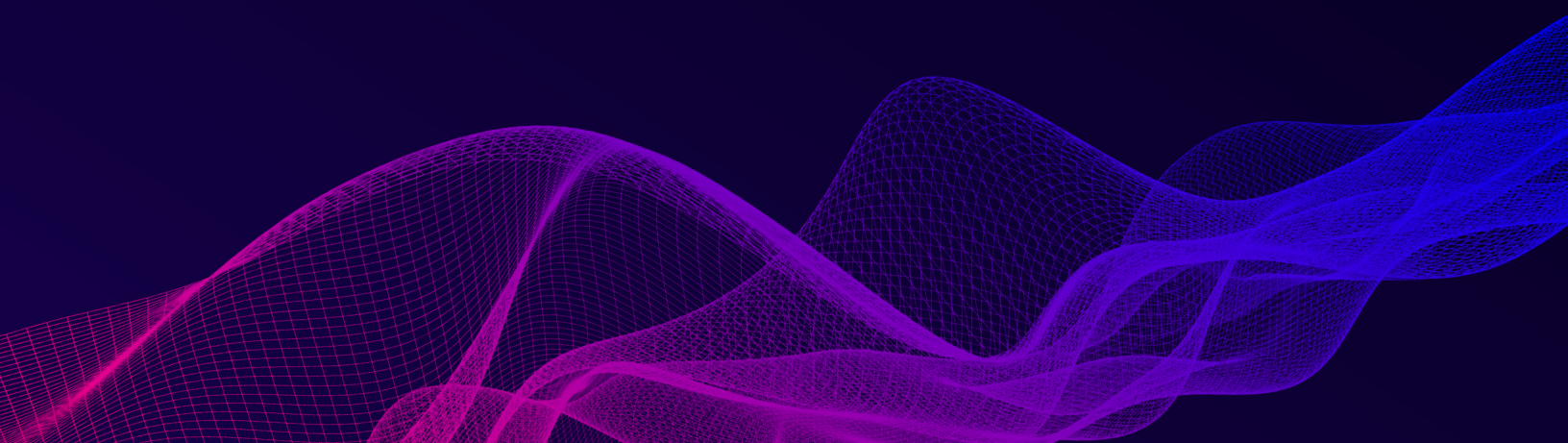
- **What:** Confirms that the vendor is meeting SLA terms related to performance, documentation, and incident reporting.
- **How:** Review vendor reports, service uptime records, update notifications, and incident communication logs; cross-check with SLA requirements.
- **Evidence:** SLA document, compliance checklist, audit reports, and correspondence with vendor representatives.

Vendor Transparency Assessment

- **What:** Measures the level of visibility the institution has into the vendor's model, data sources, and safety controls.
- **How:** Request and review vendor-provided documentation on training data governance, model update processes, and safety architecture; evaluate completeness and clarity.
- **Evidence:** Documentation packets, vendor Q&A logs, and any internal memos summarizing vendor transparency gaps.

Periodic Vendor Re-Validation

- **What:** Ensures vendor controls remain effective over time.
- **How:** Schedule and perform re-tests of guardrails, safety features, and SLA compliance at set intervals or after significant model changes.
- **Evidence:** Re-validation test results, comparison to prior results, and updated risk assessments.



9 Documentation and Audit Trail

Context & Background

In SR 11-7, documentation is not a bureaucratic afterthought—it is a core requirement for ensuring that models can be understood, explained, and reproduced. MRM teams should maintain a clear audit trail so they can trace how a system produced a given output and verify that proper controls were followed.

For agentic systems, thorough documentation is even more critical because:

- Outputs can be influenced by many factors beyond the core model, such as prompt templates, retrieval sources, fine-tuning datasets, and tool-use sequences.
- Foundation models are, by definition, non-deterministic and may also evolve over time through vendor updates, meaning that the same prompt could produce different outputs later. Without version-controlled documentation, it may be impossible to reconstruct why a given output was generated at a specific time.
- In regulated financial contexts, unresolved output discrepancies can lead to customer disputes, legal exposure, or enforcement actions.

Three key documentation concepts for agentic systems are:

- Prompt library audit – A structured review of all prompt templates and instructions used by the agent to ensure they are accurate, compliant, and up-to-date.
- Decision path reconstruction – The ability to replay or trace all inputs, retrievals, prompts, and tool calls that led to a given output.
- Version control discipline – Maintaining a complete history of all changes to prompts, retrieval corpora, configurations, and fine-tuning datasets, with timestamps and rationale.

Without these elements, institutions risk losing explainability—a violation of both SR 11-7's conceptual soundness requirements and emerging AI governance standards.

Key MRM Questions

- Is every version of fine-tuning data, prompt templates, and retrieval corpora l**ogged with version numbers, timestamps, and change rationales?**
- Can the institution reconstruct the **exact decision path** for any given output, including inputs, retrievals, prompt configuration, and tool calls?
- Is there a formal **prompt library audit process** to ensure prompts remain accurate, bias-free, and compliant?
- Are documentation and audit trails **stored securely** and protected against tampering?
- Is there a process for **periodically reviewing and updating** documentation to reflect changes in the model, vendor updates, or regulatory requirements?

Sample Tests

Prompt Library Audit

- **What:** Verifies that all prompt templates used by the agent are current, compliant, and well-documented.
- **How:** Review the full prompt library against compliance guidelines, product documentation, and intended business logic; flag outdated or non-compliant prompts.
- **Evidence:** Annotated prompt library, audit checklists, update logs, and approval records.

Decision Path Reconstruction Test

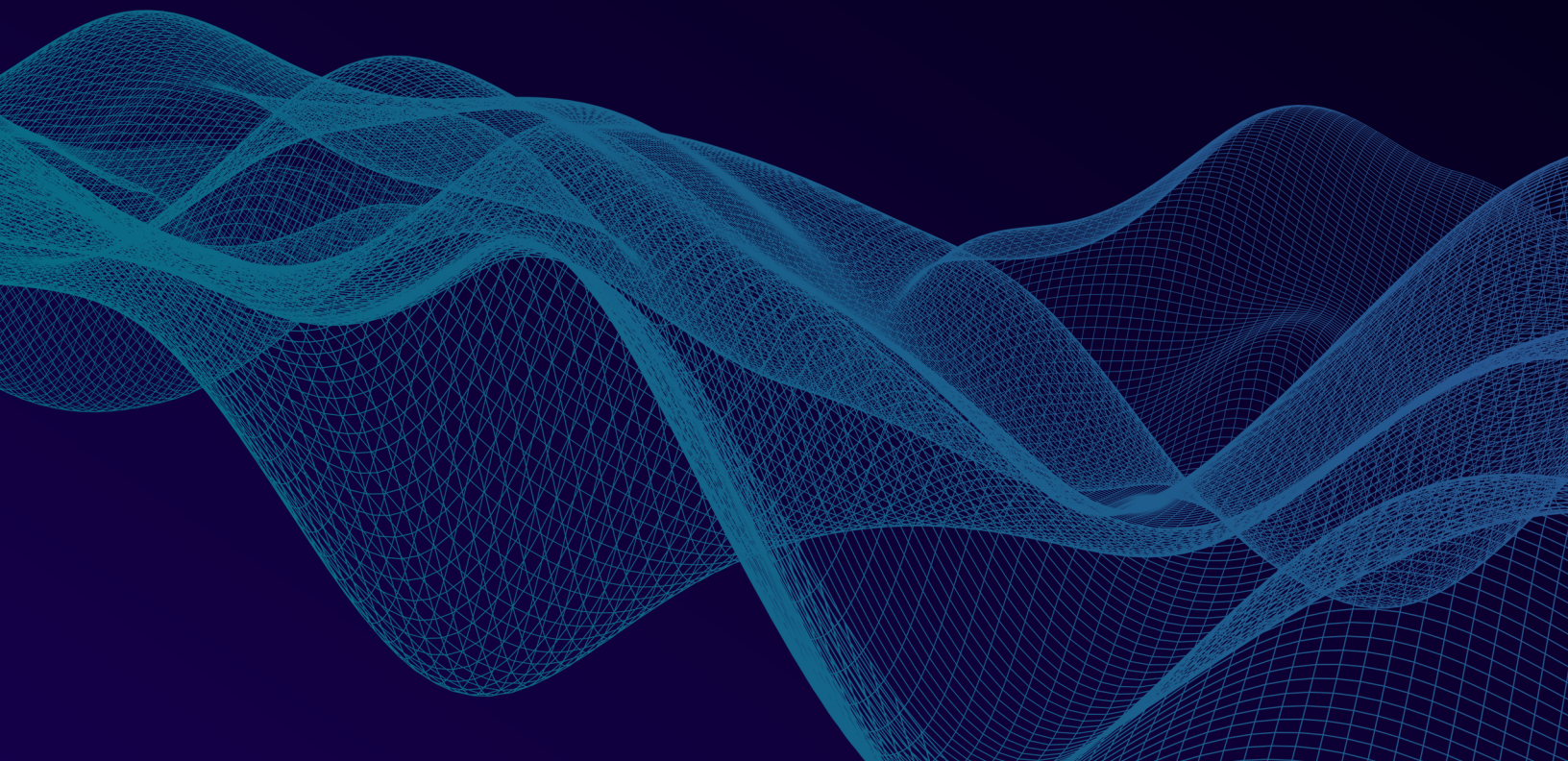
- **What:** Confirms that the institution can trace how a specific output was generated, including the intermediate reasoning steps available to the system.
- **How:** Select a historical output at random, retrieve all associated logs (inputs, retrievals, prompt version, tool calls, model version, and—where recorded—reasoning traces or intermediate outputs), and replay the process in a test environment.
- **Evidence:** Output reproduction logs, configuration files, documented chain of custody for data and model versions, and (if available) reasoning traces showing how the system arrived at its final output.

Version Control Audit

- **What:** Ensures that all changes to the agent's configuration, data, and prompts are tracked in a formal version control system.
- **How:** Compare the current configuration and datasets against the change log to verify completeness and accuracy.
- **Evidence:** Version control repository records, change tickets, approval workflows, and rollback logs.

Documentation Integrity Check

- **What:** Verifies that documentation and logs are stored securely and protected against unauthorized edits.
- **How:** Review storage location, access controls, and edit histories; attempt to retrieve historical versions for validation within reasonable timeframes.
- **Evidence:** Access control matrices, log integrity reports, and security audit findings.



10 Contingency Planning and Exit Strategy

Context & Background

Under SR 11-7, contingency planning is a key safeguard against operational and compliance breakdowns. For traditional models, this means having backup processes in place if the model fails or produces unacceptable results.

For agentic systems, the stakes are higher because:

- Dependence on vendor-hosted LLMs (e.g., ChatGPT, Claude) exposes institutions to outages, latency issues, and loss of service beyond their direct control. Outputs may shift unpredictably with vendor updates, data source changes, or prompt modifications, creating instability in system behavior.
- Many are embedded into critical workflows (e.g., customer service, loan decisioning), meaning a sudden failure could halt operations or generate widespread non-compliance.
- The “black box” nature of foundation models often makes rapid debugging difficult, so institutions need ready-to-go alternatives rather than just a plan to investigate.

Four important concepts here are:

- Fallback Systems – Any agentic use case that depends on vendor-hosted LLMs must have a tested plan for handling outages or connectivity failures. Fallback options can include reverting to pre-existing legacy systems or enabling seamless switchover to an alternative LLM provider.
- Champion–Challenger Models – The “champion” is the production model in active use, while the “challenger” is a parallel model or configuration monitored for drift, degradation, or improved performance. This setup allows for immediate replacement if the champion underperforms or fails.

- Disablement Triggers – Predefined, objective performance thresholds (e.g., factual accuracy falling below X%, hallucination rate rising above Y%) that automatically initiate fallback procedures to protect reliability and compliance.

- Parallel Monitoring – Continuously running the challenger system alongside the champion in real time, ensuring it remains production-ready and can be activated without delay if needed.

Without tested contingency planning, institutions may face dangerous incentives to keep a malfunctioning agent online—especially if it's deeply integrated into customer-facing or compliance-sensitive operations.

Key MRM Questions

- Does the institution maintain **fully functional fallback systems or workflows** that can replace the agentic system immediately if needed?

    - If fallback depends on a competing LLM, has that version been fully vetted for safe production use?
    - What predefined thresholds trigger activation of the fallback system?
    - Are fallbacks initiated automatically or do they require human approval?
    - Is redundancy built in so fallback can occur even if key personnel are unavailable?

- Are **disablement triggers** defined, objective, and tied to real-time monitoring metrics?

- Is the fallback (challenger) model or process **actively monitored** to ensure it remains production-ready?

- Has the institution **tested the switchover process** under realistic operational conditions?

- Is there an **exit strategy** for replacing or decommissioning the vendor or model if it no longer meets compliance, performance, or cost requirements?

Sample Tests

Disablement Drill

- **What:** Tests whether the institution can detect when the agent has crossed predefined risk thresholds and disable it promptly.
- **How:** Simulate a scenario where the agent's accuracy or compliance metrics fall below acceptable levels; verify that disablement triggers fire and fallback systems activate.
- **Evidence:** Incident simulation logs, monitoring dashboard screenshots, escalation records, and post-mortem reports.

Parallel Monitoring Test

- **What:** Confirms that the challenger model is kept up-to-date and capable of taking over instantly.
- **How:** Compare challenger outputs to champion outputs in real-time; verify challenger meets accuracy, fairness, and latency requirements.
- **Evidence:** Side-by-side output logs, challenger performance reports, and approval records showing readiness.

Fallback Workflow Simulation

- **What:** Validates that human-led or alternative automated workflows can replace the agent if both champion and challenger fail.
- **How:** Run a temporary shutdown of both AI systems and route tasks through the fallback workflow; measure operational continuity and compliance adherence.
- **Evidence:** Simulation results, completion time metrics, error rate analysis, and user feedback logs.

Vendor Exit Drill

- **What:** Tests the institution's ability to transition to a new vendor or model provider with minimal disruption.
- **How:** Conduct a controlled trial of onboarding a new vendor or deploying an internally developed alternative; evaluate data migration, integration, and testing steps.
- **Evidence:** Transition plan, vendor onboarding checklist, test deployment results, and lessons learned documentation.

Conclusion

The emergence of agentic systems in financial services marks a fundamental shift in how institutions deliver products, serve customers, and manage operations. While these technologies offer unprecedented flexibility and efficiency, they also introduce new categories of risk—many of which are dynamic, context-dependent, and not fully addressed by traditional model governance frameworks.

This manual adapts the well-established principles of SR 11-7 to the distinctive realities of generative and agentic AI. It outlines supplemental governance structures, validation procedures, monitoring practices, and contingency plans needed to manage these systems responsibly. It also incorporates emerging best practices from agentic governance research and insights from large-scale red teaming of deployed GenAI products.

For financial institutions, the path to safe and compliant adoption lies in embedding these controls into the full lifecycle of agentic systems—from development through deployment to retirement. Validation is not a one-time hurdle; it is an ongoing discipline that demands vigilance, documentation, and the readiness to disable or replace systems that fall short of risk tolerance or MRM standards.

By adopting the approaches outlined here—rooted in transparency, structured testing, independent review, and continuous monitoring—institutions can not only withstand regulatory scrutiny but also build resilient, trustworthy agentic systems that enhance customer trust, support long-term strategic goals, and uphold the integrity of the financial system.

If your institution is preparing an agentic system for MRM review, an independent third-party validation can provide added assurance that governance, testing, and documentation meet the highest standards. FairPlay conducts independent validation of agentic systems for financial institutions and insurance carriers. To learn more, contact **info@fairplay.ai.**

fairplay