Reliability and Repair for Agentic Systems

Reins AI Technical White Paper v1.0, October 2025

Authors

Marisa Ferrara Boston, PhD
Managing Partner, Reins AI

Heather Frase, PhD
Evaluation Lead, Reins AI
CEO, VerAITech

Effi Georgala, PhD
Agent Reliability Lead, Reins AI
CEO, Metron AI

Abstract

Artificial intelligence is moving from experimental pilots to embedded infrastructure across regulated domains such as audit, finance, and professional services. As these systems begin to make or influence decisions that carry strategic, financial, and reputational risk, their reliability can no longer be assured by static validation alone. This white paper presents a framework for *Reliability & Repair*: a structured, repeatable process for detecting, triaging, simulating, repairing, and verifying failures in complex AI systems. By combining established reliability-engineering practices with modern AI monitoring techniques, it demonstrates how organizations can measure reliability growth, align risk with severity, and transition from passive oversight to continuous improvement.

Keywords:

Reliability growth, AI assurance, agentic systems, monitoring, repair, risk alignment

---

Please cite as:

Boston, M. F., Frase, H., & Georgala, E. (2025). *Reliability and Repair for Agentic Systems.* Reins AI Technical White Paper v1.0. October 2025. Retrieved from www.reinsai.com/articles/reliability-and-repair-for-agentic-systems

1. Introduction: Why Reliability Matters Now

Artificial intelligence is moving from experimental pilots to embedded infrastructure across regulated domains such as audit, finance, and professional services. As these systems begin to make or influence decisions that carry strategic, financial and reputational risk, the question is no longer "Does the model work?" but "When does the system remain reliable and when doesn't it?"

Traditional assurance methods focus on validation at deployment: benchmark accuracy, policy compliance, or red-team testing. Yet once deployed, AI systems become dynamic, adaptive, and interdependent. Their reliability cannot be guaranteed by static testing alone. Failures emerge not just from model behavior, but from integration points, context drift, and unforeseen user interactions.

In audit and finance workflows, these failures could manifest as mis-classified transactions, incomplete evidence gathering, or inconsistent reasoning chains, errors that undermine both efficiency and trust. Reliability, long treated as a hardware or safety-engineering concern, must now extend to agentic and cognitive systems that learn, interact, and evolve.

This white paper presents a framework for Reliability & Repair: a structured process for detecting, triaging, reproducing, repairing, and verifying failures in complex AI systems. Drawing from established reliability engineering and emerging monitoring practices for large language-model systems, we demonstrate how reliability growth can be quantified, how risk can be aligned with severity, and how continuous repair transforms monitoring from a passive dashboard into an active improvement loop.

2. The Problem: Current State of Agent Reliability

Current AI-monitoring approaches treat failure as an endpoint. Dashboards flag anomalies, precision metrics drop, alerts fire, but the process often stops there. Failures are counted, not cured. Mean Time to Repair remains long; oversight is diffuse; and remediation actions are rarely captured or measured for effectiveness.

Three structural gaps dominate today's reliability landscape:

1. Risk Misalignment. Failures are tracked by frequency, not by *consequence*. Minor deviations and catastrophic breakdowns are reported in the same units, obscuring which failures truly elevate audit or business risk.
2. Lack of a Repair Loop. Monitoring pipelines detect and classify but rarely close the loop to simulation, correction, and verification. Without a feedback path, reliability data accumulates without yielding reliability growth.
3. Fragmented Oversight. Human review remains essential, but is often applied uniformly rather than strategically. Oversight resources are spent on low-impact anomalies while high-severity incidents slip through delayed or unnoticed.

As a result, organizations have visibility into *what went wrong*, but not a repeatable process for *making it right*. True reliability requires a transition from detection to repair. From static measurement to dynamic improvement.

It is critical to note that AI-enabled system-level reliability differs fundamentally from AI model-level or agent-level reliability. Model-level and agent-level reliability typically focus on performance metrics such as accuracy, in-distribution performance, and uncertainty estimation for individual models or agents. In contrast, system-level reliability does not evaluate AI models or agents in isolation, but rather examines their integration within the broader system to assess overall system performance when AI components fail.

System-level reliability emphasizes operational impact: measuring the severity of system failure outcomes and the extent to which failures impede the system's ability to perform its intended application or task. This system-level perspective becomes particularly critical in multi-agent architectures where multiple AI agents must coordinate and collaborate effectively to achieve system objectives. This work, and the monitoring framework we've developed, is entirely focused on the latter.

3. The Framework: Reliability & Repair as the Missing Link

This section provides the foundational concepts of reliability growth, a process for tracking and improving reliability. It will also discuss its relation to failure severity and risk. Later sections will discuss using formal reliability processes for complex agentic-AI systems.

3.1 What is Reliability?

In systems engineering Reliability has a specific and longstanding definition[1]. It is:

Reliability: The ability of a system to

- Perform its intended task
- At a minimal, acceptable level of quality
- Without failure
- For a stated amount of [time / distance / cycles / computations]

3.2 Reliability Growth: a Path to Mature and Stable Systems

Reliability analysis and assessment practices are employed across sectors, including the U.S. Department of Defense (DoD),[2] software companies, and manufacturers[3]. Organizations also use them to assess a product's achieved and potential reliability. They facilitate identifying failure modes and prioritizing the most impactful ones. Reliability Growth analysis is one of many reliability analytics and metrics.

By tracking reliability data, failures, and failure modes, an organization can also understand a system's development stage, current reliability, and potential reliability. A system's reliability changes as the corrective actions for failure modes, design changes, and maintenance or new operational procedures are implemented. These changes in reliability data over time can create a curve–a reliability growth curve.

A system's reliability data is often fitted to one of a handful of established reliability growth models. There are many models for reliability growth, with two prominent models being the Duane and the Crow-AMSAA models. Observations[4] by J. T. Duane, often cited as the originator of reliability growth curves, led to an observation-based or empirical model[5] where the cumulative Mean Time Between Failure (MTBF, with

---

[1] Marvin Rausand and Arnljot Høyland. System Reliability Theory: Models, Statistical Methods and Applications. Wiley-Interscience, Hoboken, NJ, 2004.
[2] See https://www.dau.edu/acquipedia-article/reliability-growth https://www.dote.osd.mil/Portals/97/docs/TEMPGuide/Reliability_Growth_Guidance_3.0.pdf?ver=2019-08-26-165237-870, and https://nap.nationalacademies.org/read/18987/chapter/2#5.
[3] See https://www.ge.com/digital/documentation/meridium/V36160/Help/Master/Subsystems/Reliability/Reliability.htm#What_is_a_Reliability_Growth_Analysis_.htm
[4] Duane, J.T., "Learning Curve Approach To Reliability Monitoring," IEEE Transactions on Aerospace, Vol. 2, pp. 563-566, 1964.
[5] See https://help.reliasoft.com/reference/reliability_growth_and_repairable_system_analysis/rg_rsa/duane_model.html

failure rate = 1/MTBF) has a logarithmic behavior. Another frequently used model is Crow-AMSAA[6]. While the Crow-AMSAA model builds upon Duane's work, it is, however, a statistics-based vs observation-based model. Additionally, there are variants of the Crow model which incorporate the different ways systems are tested, repaired, and maintained. The Crow-AMSAA model, which DoD formalized in MIL-HDBK-189[7], is a statistical extension of the earlier empirical Duane model, and both are widely used across engineering sectors.

Reliability Growth Curves are a common and powerful systems engineering tool for understanding a system's maturation throughout its development and operational lifecycle. These curves track a reliability metric as the system is used over time (see Figure 1). Typically, during early development stages, the growth curve shows a rapid improvement as major and easily identifiable issues are corrected. As the system matures, the curve becomes flatter. The few remaining failure modes are understood, but there are few corrective actions implemented because the remaining ones are difficult or costly. In this phase, the system's reliability may or (unfortunately) may not meet its goals or requirements. If the system needs additional reliability improvement, it may require a major technological change–or a different system entirely. In this flat part of the reliability growth curve, which indicates a mature, stable system, alerts based on both organizational requirements- and statistical-based thresholds work effectively. Additionally, performance optimization is more effective during a system's mature phase.

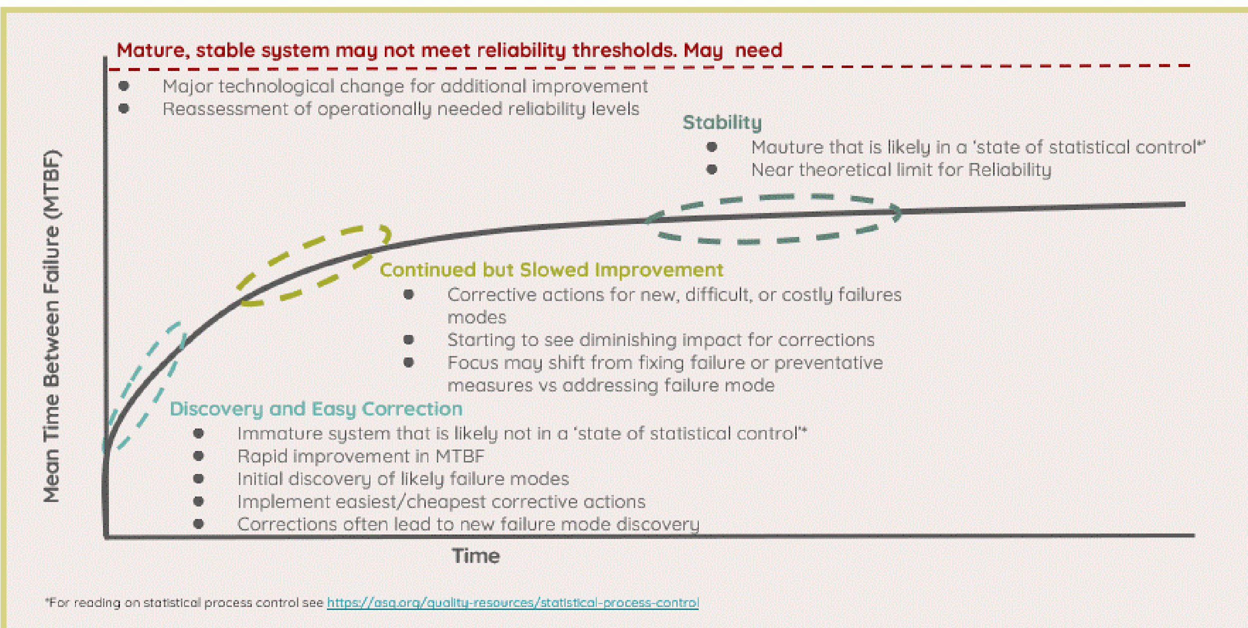

Figure 1: Theoretical Reliability Curve. Phases of a Basic and Notional Reliability Growth Curve.

While MTBF is the most common metric used for monitoring reliability growth, other metrics can be used. Often, time is not a relevant unit for a system. Distance (mean miles between failures) and events (mean events between failures) are also common. Additionally, sometimes the failure rate (failure rate = 1/MTBF) is tracked instead of MTBF.

[6] Developed at the U.S. Army Materiel Systems Analysis Activity.
[7] MIL-HDBK-189. 13 February 1981. Department of Defense. Handbook. Reliability Growth Management. https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=53928

3.3 Severity: Some Failures Matter More

When tracking reliability growth, you generally do not use all failures to calculate MTBF. This is because many failures do not impact a system's ability to perform its intended task at a minimum level of quality (Section 3.1). Typically, only the more severe failures are used for calculating reliability[8]. Thus establishing failure severity levels and definitions is a key part of monitoring and improving reliability.

Context-relevant and use-specific failure severity levels are fundamental to meaningful reliability growth tracking because they bridge the gap between abstract technical metrics and organizationally-relevant operational impact. Traditional reliability analysis often employs generic severity classifications that fail to account for how identical failure modes can manifest vastly different consequences across different operational contexts and user environments. For example, a 5-minute processing time might represent a minor inconvenience in academic research, but could constitute a critical failure for a medical diagnostic system. By developing severity frameworks that reflect actual business impact, organizations can more accurately model reliability growth trajectories and prioritize corrective actions.

A failure's severity level can be defined in multiple ways. Common approaches include assessing the harshness of a failure's consequences on entities (people, organizations, places, etc.) or determining how significantly the system's mission or tasks were affected. Regardless of the approach (or approaches) chosen, the severity scoring criteria need to be clear, detailed, and documented.

For the military, MIL-STD-882E[9] defines severity as the magnitude (i.e. harshness) of potential consequences of a mishap (i.e. failure). Additionally, the document provides definitions for four severity levels (negligible, marginal, critical, and catastrophic). The definitions in MIL-STD-882E are limited, but they have a history of being tailored[10] and could be adapted to consider social implications. Table 1 shows examples of scoring criteria for different severity categories. Note that the criteria can vary with the type of entity being impacted.

[8] For example, your car's burnt out dome light is a minor failure that does not impact its reliability, but a broken axis does.
[9] MIL-STD-882E, Department of Defense Standard Practice: System Safety, 11 May 2012, https://safety.army.mil/Portals/0/Documents/ON-DUTY/SYSTEMSAFETY/Standard/MIL-STD-882E-change-1.pdf
[10] https://apps.dtic.mil/sti/pdfs/ADA619377.pdf

Description	Severity Category	Failure Result Criteria Examples		
		Individual Health Focused	Environment Focused	Financial Loss Focused
Catastrophic	I	Death or permanent total disability	Irreversible significant environmental impact	Monetary loss (or equivalent property damage) equal to or exceeding $10M
Critical	II	Permanent partial disability, injuries, or occupational illness that may result in the hospitalization of at least three personnel	Reversible significant environmental impact	Monetary loss (or equivalent property damage) equal to or exceeding $1M but less than$10M
Marginal	III	Injury or occupational illness resulting in one or more lost work day(s)	Reversible moderate environmental impact	Monetary loss (or equivalent property damage) equal to or exceeding $100K but less than $1M
Negligible	IV	Injury or occupational illness not resulting in a lost workday	Minimal environmental impact	Monetary loss (or equivalent property damage) less than $100K
Severity criteria from MIL-STD-882E				

Table 1: Harm Severity. Example of Scoring Criteria for Severity Categories Based on Entity Impact.

Severity that focuses on operations captures the degree to which a failure impacts the user's ability to complete a mission or task. Table 2 has examples of operations-based failure severity. For systems that are expected to have continuous operations, classification as severe failures may be related to the amount of downtime or repair effort.[11] [12] Severe mission based failures may also be those that result in inaccurate results or poor system performance.

[11] https://nij.ojp.gov/sites/g/files/xyckuh171/files/media/document/draft-failure-definitions-and-scoring-criteria.docx

[12] FEMA has identified different function levels for its operations: Primary MIssion Essential Functions, Mission Essential Function, and Essential Supporting Activities. For FEMA, a failure's severity level could be determined by which of these function levels was impacted. https://www.fema.gov/sites/default/files/2020-07/Federal_Continuity_Directive-2_June132017.pdf

Severity Category	Failure Result Criteria Examples		
	Airborne Radar	Continuously Operating AI	Personal Vehicle
1	Engine failure prevents safe flight.	Image generators produce child sexual abuse material (CSAM).	A tire blows out and needs to be replaced.
2	Some radar antenna elements are not working. The radar is operable, but its performance is degraded.	Image generators cannot consistently remove types of objects (e.g., dogs, airplanes, cars, etc.) when requested through a text prompt.	The internal GPS navigation system has an old map and needs updating. The system usually works well, but more recent maps would prevent wrong or missed turns.
3	An overhead interior light needs replacing, but operations are not impacted.	Created images sometimes have hands with 6 fingers.	A small dent in the passenger door.

Table 2: Example of Scoring Criteria for Severity Categories Based on Mission/Task Impact.

3.4 Relationship Between Risk, Severity, and Reliability

Risk, severity, and reliability are interconnected. Risk is the probability of a *specific* failure (sometimes called a hazard for a failure) multiplied by the severity of a failure. Reliability provides information about how frequent *any* failure of above a selective severity level (usually focusing on critical or catastrophic failures) occurs. Thus both risk and reliability contain some information about failure likelihood and severity. However, risk tends to consider specific failures individually, assigning separate risk values to each failure type or failure mode. Reliability is more focused on system operations as a whole and how often it will have failures of unwanted severity levels. Figure 2 provides a visual representation of the relationships. While risk score is commonly at the failure level and reliability at the system level, there are approaches to aggregating and disaggregating (respectively) these metrics.

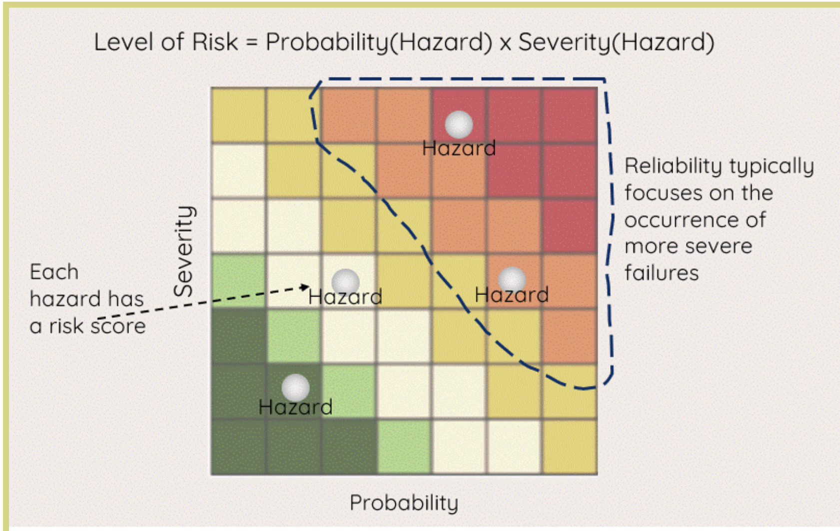

Figure 2: Phases of a Basic and Notional Reliability Growth Curve

4. The Solution: From Failures to Repair Packets to Measurable Improvement

Reliability in agentic systems cannot be achieved by detection alone. Dashboards, alerts, and static monitoring can identify that failures occur, but they do not prescribe what to change or how to improve. To reduce risk in high-stakes domains like audit and finance, we need a structured loop that turns failures into repair packets: concrete, testable improvements that increase system reliability over time.

This section outlines a five-part process, Monitor, Triage, Simulate, Repair, and Verify, that extends beyond traditional Quality Control (see Figure 3).

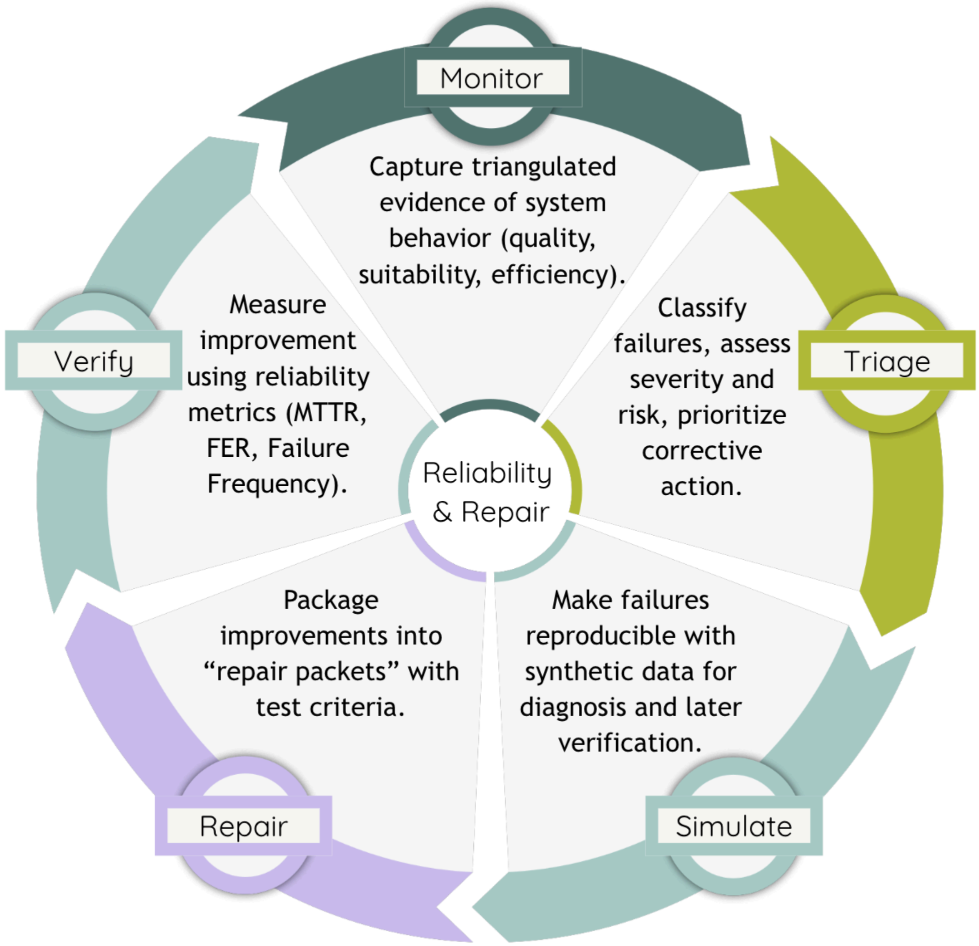

Figure 3: The typical improvement loop finds new potential within agentic systems as LLMs provide opportunities to observe and automate key aspects of complex behavior that were previously too costly or difficult to implement.

- Monitoring generates triangulated evidence of failures across traces, outputs, and interactions.
- Triaging converts detection into prescription by clustering incidents, calibrating severity and risk, and prioritizing what to correct first.
- Simulating recreates failure conditions with synthetic case signatures and scenario specs so issues are reproducible, representative, and safe to test *without relying on client data*.
- Repairing packages each prioritized issue into an actionable improvement unit (a repair packet), containing the scenario, hypothesis, targeted adaptation, acceptance tests, and representative synthetic data.
- Verifying runs before and after tests on a simulation bench and in controlled canary environments to quantify impact using reliability metrics and calibrated human-oversight outcomes.

Together, these steps create a repeatable loop that allows agentic monitoring systems not just to detect failures, but to *improve* through them, so that we can shorten Mean Time To Repair, reduce critical incidents, and align reliability with business risk.

4.1 Monitor

Reliability begins with monitoring. Without continuous, structured observation of agent behavior, there is no foundation for improvement, and no evidence for mitigating risk. Monitoring is not just the first step in the loop, it grounds every other step in the process toward creating reliable systems.

Often agentic systems are immature, they are unstable and prone to frequent failure with a large variety of root-causes or failure modes. Dashboards for immature systems have limited utility. Dashboards are more valuable for monitoring operational systems where visibility can highlight issues and facilitate rapid mitigation. However, in unreliable systems with varied failure patterns, dashboards often become counterproductive, generating continuous alerts without providing clear paths to resolution. Additionally, when failure rates are high, the dashboard essentially becomes a real-time failure log rather than a proactive tool.

Because agentic systems can be unreliable and more akin to complex systems than traditional software, dashboards and single metrics are insufficient. Failures must be captured through triangulated evidence: traces of agent behavior, human guidance and interventions, and final outputs. Together, these streams expose not only *whether* a failure occurred, but help us automatically triage *where, why,* and *with what consequence* (discussed in the next section).

Figure 4 diagrams our typical monitoring suite. The suite takes observations from the human-agentic interactions and deliverables and runs automatic evaluators (either rule-based, statistical, LLM-based, or combined) to determine the overall system performance. Performance for our purposes is bucketed into the following categories:

- Suitability
    - Assesses whether the agentic system is delivering helpful, relevant guidance to users.
    - Typical evaluators include
        - Interaction assessments
        - Human intervention assessments (positive and negative)
        - Memory failures
        - Fallback responses
        - Engagement drop-off
- Quality
    - Focuses on the accuracy and acceptability of AI outputs
    - Typical evaluators include
        - Error checks
        - Hallucination checks
        - Information verification
        - Output accuracy
        - Output completeness
        - Output relevancy
- Efficiency
    - Compares overall task and review timing to determine whether the complex system is providing value across the overall stream.
    - Typical evaluators include

- Task timing measures
- Latency measures
- Cost measures

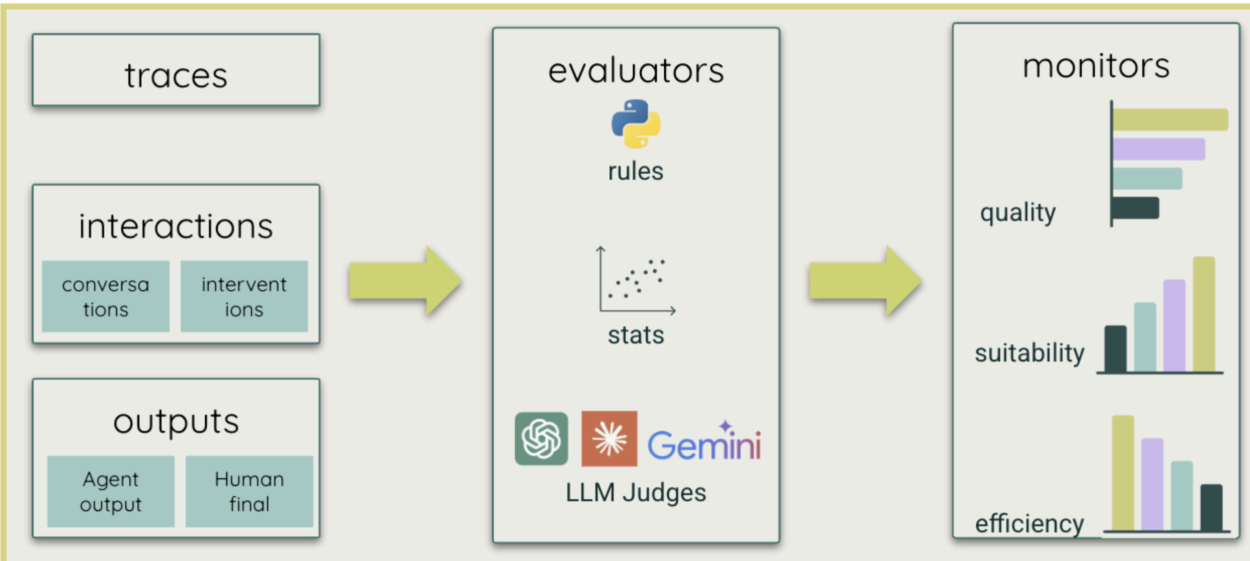

Figure 4: With agentic telemetry and targeted LLM judges, we achieve triangulated evidence of failures and successes through *quality, suitability,* and *efficiency*, not just single thresholds.

Monitoring systems designed with these objectives highlight *variability* and severe failure rather than *mean performance*. Systems that perform well on average but fail catastrophically in edge cases pose the highest operational risk. In our early deployments, high variance in instruction adherence and deliverable accuracy was far more damaging to trust than mean performance scores indicated. Monitoring makes these extremes visible so they can be acted upon.

A final critical design element for our monitoring systems is that they be *active*, not *archival*. Every captured signal flows into triage, simulation, and repair pipelines, ensuring that production data becomes the basis for structured reliability growth. In this approach, monitoring is not a dashboard, but the backbone for improvement, operational management, and risk mitigation.

4.2 Triage

Triaging converts detection into prescription by clustering incidents, calibrating severity and risk, and prioritizing what to correct first. Effective triage draws upon established reliability engineering practices, particularly concepts from Failure Mode and Effects Analysis (FMEA)[13] [14] [15], to ensure systematic and

[13] IEC 60812:2018, Failure modes and effects analysis (FMEA and FMECA), 10 August, 2018.
[14] *Procedures for Performing a Failure Mode, Effects and Criticality Analysis. A. U.S. Department of Defense.* 1980. MIL–HDBK–1629A.
https://web.archive.org/web/20110722222459/https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=37027
[15] Stamatis, DH (1995) Failure Mode and Effect Analysis: FMEA from Theory to Execution. ASQC Quality Press, Milwaukee, WI.

rigorous failure assessment. A well implemented triage enables teams to focus on the most critical issues first and allocate resources effectively.

Our Reliability and Repair system's triage process consists of four primary phases (see Figure 5):

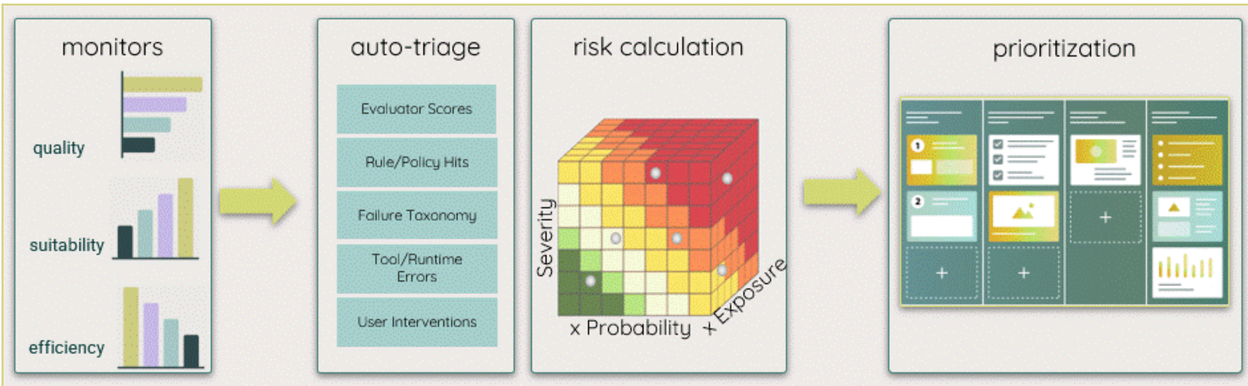

Figure 5: Triangulated evidence from system monitors feeds into automated processes for triage, risk classification, and prioritization. Triage clusters, risk ranks, humans confirm.

1. Failure Classification Failures are categorized by type and clustered into failure classes that require attention. Classification enables pattern recognition across similar failures and helps identify whether failures are isolated incidents or symptoms of systemic issues.

2. Risk Classification Classified failures undergo risk assessment, which calibrates risk levels based on multiple factors which may include:

- Severity: The consequence or impact of the failure on system operations, users, and organizational objectives
- Probability: The likelihood that the failure will occur or recur
- Scope/Exposure (optional 3rd axis)[16]: The breadth of impact, including operational scope across different sectors or contexts, the range of system components affected, or the duration of the failure's effects
- Organizational Impact: Broader consequences such as regulatory compliance issues, reputational risk, and financial costs

This multi-dimensional risk classification draws from FMEA methodology, which provides a systematic framework for assessing failure risk across multiple factors. While traditional FMEA focuses on severity, occurrence, and detection, this triage approach expands the risk assessment to capture the broader operational and organizational context. Regardless of the specific approach, the goal is to have a systematic, repeatable method that supports objective prioritization.

[16] All risk cubes have "Probability" and "Severity" axes, although slightly different names (e.g., "Likelihood" and "Consequence") may be used. When just these two axes are used, the term risk matrix instead of risk cube is usually used. If an optional 3rd axis is used, it is then always called a risk cube. The quality on the third axis varies depending upon the sector or application.

3. Failure Prioritization Risk-classified failures are ranked according to their urgency and importance. Prioritization balances multiple considerations:

- Risk levels determined in the previous phase
- Available resources and technical expertise
- Dependencies between failures and potential cascading effects

Failure prioritization identifies which failures require immediate attention and which can be addressed later. The process acts as a filter, ensuring that the most important failures advance to resource-intensive investigation phases.

4. Investigation and Response Determination Teams analyze the prioritized failures, identify root causes, and determine appropriate response types. This phase mirrors FMEA's corrective action development, where understanding the root cause enables targeted interventions. In our Reliability and Repair system, the response types include:

- Corrective Actions: Fundamental improvements that address root causes, enhance system reliability, and prevent future failures
- Workarounds: Process or operational changes that maintain system functionality without addressing underlying technical issues
- Patches: Temporary technical fixes that mitigate immediate problems while more comprehensive solutions are developed. Typically, patches restore system operations but have minimal impact on reliability.

Response determination considers not only the technical nature of the failure but also implementation feasibility, resource requirements, and alignment with organizational priorities. The investigation phase may reveal that a single root cause underlies multiple failure classes, enabling more efficient and effective interventions.

This comprehensive triage approach ensures failures are systematically assessed, prioritized based on risk, and matched with responses appropriate to their nature and severity. Through FMEA-based multi-factor risk assessment and structured prioritization, triage converts detection findings into prioritized responses. These responses guide effective system repair and reliability improvement.

4.3 Simulate

Once failures are detected and triaged, the next challenge is making them reproducible. Reproducibility is critical because it allows us to:

- Verify corrective actions
    - Ensure that the proposed correction actually resolves the issue without introducing new problems.
- Enable root cause analysis
    - By recreating the failure conditions, engineers can inspect logs, inputs, and system states to isolate the underlying cause.

- Facilitate consistent evaluation
    - Reproducible failures can be re-run across versions or environments to test regression and resilience.
- Support knowledge transfer
    - Capturing reproducible examples creates training data and documentation for future operators, improving organizational memory.
- Improve trust and governance
    - When stakeholders can independently reproduce failures and confirm corrections, it strengthens confidence in the system's reliability process.

Real-world data is often protected, inconsistent, or too narrow to serve as a reliable testbed. To repair effectively, we need to recreate failure conditions in controlled environments. This is where simulation becomes a critical part of the process. The addition of generative AI allows this once costly and difficult process to be not only feasible, but observable and intuitive.

We start by extracting a *case signature* (the pattern of a failure) paired with a *scenario specification* (describing the context in which it occurred) (Figure 6). From this foundation we generate synthetic documents with data drawn from fictional but realistic companies. These synthetic assets are not copies of client data, but tailored recreations designed to mirror the structure, complexity, and stressors that triggered the original failure. The simulations also allow us to generate innumerable variants of the failure to help expand the use case and hone the potential repairs needed.

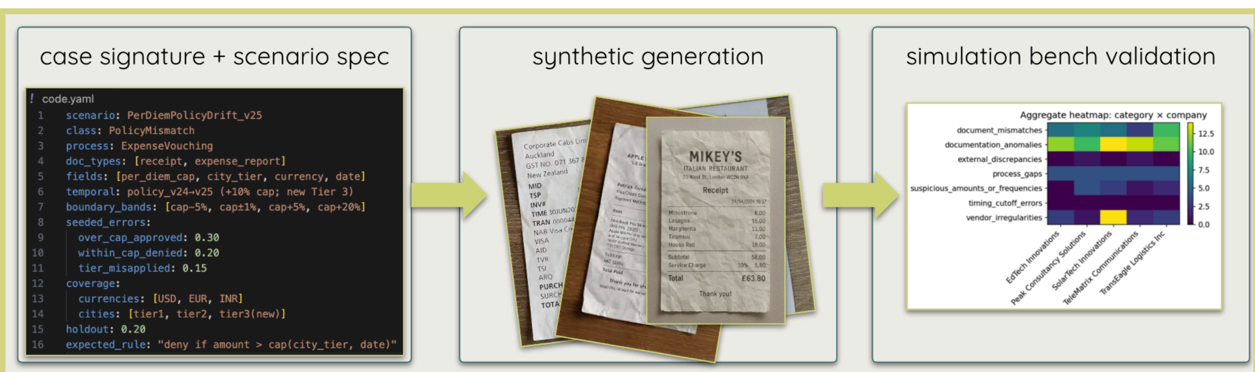

Figure 6: Synthetic data recreates failure conditions, not client data. Realism and fitness-for-use are validated through simulated agents and synthetic benches.

The cases and synthetic data are validated against a *simulation bench*: a controlled environment where agents (either client or our own replicas) can be run repeatedly under the same conditions. The bench ensures not only that the failures are reproducible, testable, and comparable across iterations, but also whether the synthetic cases are realistic enough to the real-world tasks.

Using simulations has three key advantages over traditional approaches that attempt to deidentify or reuse the original failure data:

- Safety: Simulations eliminate the need to expose or recycle sensitive client data

- Repeatability: Simulations transform one-off incidents into standardized test cases that can be re-run with every system update
- Fitness for use: Simulations allow us to validate whether a repair actually resolves the failure under representative conditions

By embedding simulation into the repair loop, failures become evidence-backed scenarios that drive measurable improvement. This recreation step transforms raw monitoring signals into the building blocks of repair packets.

4.4 Repair

Repair turns a prioritized failure into a *testable, durable improvement*. Each change is documented and verified so reliability gains are cumulative, trackable, auditable, and resistant to regression.

A repair packet (a framework repair activity) allows for the systematization and automation of repair processes. Each high-priority issue should have a repair packet, a concise unit of corrective action. A repair packet includes:

- The failure description and scenario signature
- A root-cause hypothesis
- The proposed adaptation or system adjustment
- Acceptance criteria and evaluation tests that define "done"
- Any required human validation

Over time, these packets build a *knowledge base of system repairs* (what failed, why, how we fixed it, and how we proved it), making improvements cumulative and traceable. By collecting failures, failure signatures, failure modes, root-cause analysis, and mitigations, knowledge can be transferred to repair and reliability processes for other systems.

In situations where failure signatures are stable and well understood, parts of the loop can run automatically under explicit safety bounds. For example, switching to a fallback retrieval strategy when evidence-grounding score confidence falls below the defined control-limit, or auto-escalating to human review when risk exceeds a threshold. These automations use the same acceptance criteria as manual fixes and leave an audit-ready trail so actions can be reviewed and, if needed, rolled back. Self-healing[17] speeds response without bypassing governance; it augments, rather than replaces, the repair-packet process.

Repair converts evidence into measurable reliability growth: scoped changes that are risk-aligned, auditable, and regression-protected, setting up Verify to close the loop.

17 https://aithority.com/machine-learning/self-healing-ai-systems-how-autonomous-ai-agents-detect-prevent-and-fix-operational-failures/#:~:text=paramount.%20Self,and%20improve%20overall%20system%20efficiency

4.5 Verify

Repair without verification is guesswork. To demonstrate true reliability growth, every change must be tested, measured, and published. Verification is the stage where improvement becomes evidence for the next cycle of adaptations.

Each repair packet is run against a *simulation bench* and, where safe, *controlled canaries in production*. We measure before/after performance across a defined set of reliability metrics:

- Mean Time To Repair (MTTR)
  - How quickly failures are identified, reproduced, and resolved.
  - This should go down as the process becomes standardized.
- Fix Effectiveness Rate (FER)
  - Percentage of repairs that successfully eliminate the targeted failure mode
  - Calculation: (Number of corrective actions that resolved the failure mode / Total corrective actions implemented) × 100%
  - Tracks if the fix actually worked as intended.
- Failure Frequency
  - How often severe failures occur in production.
  - We expect this to decrease within versions (although it can increase between significant version changes).
- Calibrated Human Oversight
  - Whether the right failures are escalated to the right humans at the right time.
  - We expect this to increase with operational maturity based on the monitoring data.

The output of the verification process is a *manifest*:

- Manifest
  - What was tested
  - What was improved
  - What risks remain

Once created, the manifest is published and fed into dashboards to create an audit trail for system maturation. This is essential for regulated domains, providing not just technical validation, but defensible documentation that the system is improving in a structured, measurable way.

Together, the verification process and the manifest make reliability improvements auditable and usable. They ensure every repair is not only tested, but recorded, communicated, and tied back to business and audit risk. This establishes agentic system progress not as promises, but as published evidence of reliability growth.

5. Reliability & Repair for Operational Recommendations

Reliability & Repair outcomes are not just technical artifacts. They directly inform how organizations should design oversight. Reporting turns system metrics into operational guidance, showing where humans must remain in the loop and where automation can safely take over.

5.1 From Metrics to Oversight

The published manifest (see Section 4.5) provides the basis for oversight design. From it, we derive:

- Reliability Growth Curves, which show whether the system is maturing or plateauing. These curves help leaders decide when automation can be trusted with greater autonomy.
- Control Charts, which expose when error rates fall outside of statistically valid thresholds. These provide triggers for human review or escalation.
- Oversight Manifests, which document which classes of failures have been eliminated, which remain rare but possible, and which still require routine human attention.

5.1 From Over-Supervision to Right-Sized Design

By aligning oversight to reliability evidence, organizations avoid two extremes: “over-supervision” that slows operations and breeds mistrust of automation, and “under-supervision” that lets severe risks escape unchecked. The goal is right-sized human oversight, supported by transparent evidence that shows where intervention adds value and where it does not.

Lessons from safety-critical industries reinforce this approach. In military decision-making, aerospace, and nuclear operations, systems are designed so that humans intervene at *critical junctions*, while automation manages routine execution. This follows a few key principles:

- Right place, right time. Oversights must be targeted, not universal. Humans should remain in the loop for *high-severity, high-impact failures*, but not for minor anomalies that are well-controlled.
- Calibrated by severity and impact. Oversight should scale with risk: catastrophic failures demand proactive human review; marginal anomalies may only warrant retrospective sampling.

Reliability & Repair does not dictate the specific protocols for oversight, but it does provide the foundational evidence that enables organizations to design them. With manifest data, growth curves, and control charts in hand, teams can establish oversight practices that are both effective and efficient.

6. Conclusion & Call to Action

The Reliability & Repair framework transforms AI assurance from reactive governance to continuous improvement. When failures are monitored, triaged, simulated, repaired, and verified in structured loops, systems not only recover faster but also become demonstrably safer and more predictable over time.

The metrics that matter: Mean Time to Repair, Fix Effectiveness Rate, Failure Frequency, and Calibrated Human Oversight form the quantitative backbone of that growth. Together they provide defensible evidence that complex agentic systems can mature just as physical and software systems have before them: through measured reliability improvement.

For technical teams, this framework offers a bridge between reliability engineering and modern AI operations. For governance and risk stakeholders , it offers a method to align oversight with evidence rather than intuition. The challenge ahead is not simply to monitor intelligent systems, but to design them to learn from failure safely.

Reliability and repair make that learning process observable, auditable, and ultimately trustworthy: the foundation upon which safe scale and resilient automation will depend.

7. Definitions

7.1 Risk

In the context of agentic and AI-enabled systems, risk represents the *potential consequence of failure*: the combined effect of how likely a failure is to occur and how severe its impact will be on the system's intended mission or organizational objectives.

Formally, it can be expressed as:

Risk = Probability of Failure × Severity of Consequence

However, in complex, adaptive systems, this simple equation should be interpreted through a *system-level lens*. Risk is not confined to individual model errors or isolated agent behaviors; it also arises from the interactions among agents, users, and environments that amplify or mitigate those failures. Additionally, a low-probability event in one subsystem may become high-risk when it cascades across dependent components, has a severe impact upon system operations, or erodes human trust.

From a reliability-growth perspective, risk serves two key purposes:

1. Calibration of Priorities. By quantifying both likelihood and severity, teams can distinguish between failures that are operationally negligible and those that are critical to safety, audit integrity, or business continuity.

2. Measurement of Progress. As reliability improves and failure frequencies decrease, residual risk should decline proportionally, especially for high-severity categories.

In audit and assurance applications, *risk-aligned reliability* means that metrics of system performance (Mean Time to Repair, Fix Effectiveness Rate, Failure Frequency) are interpreted not in isolation, but in terms of how effectively the system prevents or contains failures that could compromise assurance quality, compliance, or public trust.

7.2 Reliability: Theory Versus Reality

In practice, reliability growth curves rarely look like the one in Figure 1, but often look like the red line in Figure 7. Theoretical reliability growth curves are smooth because they assume that failure modes are continuously corrected. In reality, corrective actions are often done in spurts, with many being implemented at once, like an OS update that repairs multiple bugs and improvements in a single iteration.[18]

Systems often do not meet their desired level of reliability. When this happens the reliability growth curve fails to have sufficient growth and never crosses the reliability threshold or desired reliability level. This may happen for a variety of reasons, including

1. A system is still immature and needs more failure mode corrections
2. Incomplete Failure Mode and Effects Analysis (FMEA) leading to unidentified failure modes
3. A system is mature, but additional improvements would require a fundamental technology or structural change
4. There are too many single points of failure
5. The reliability threshold was poorly established, unrealistic, or misaligned with operational needs
6. Integration with users or other systems are introducing failures that are not contained within the system and need to be addressed at the system-of-system level

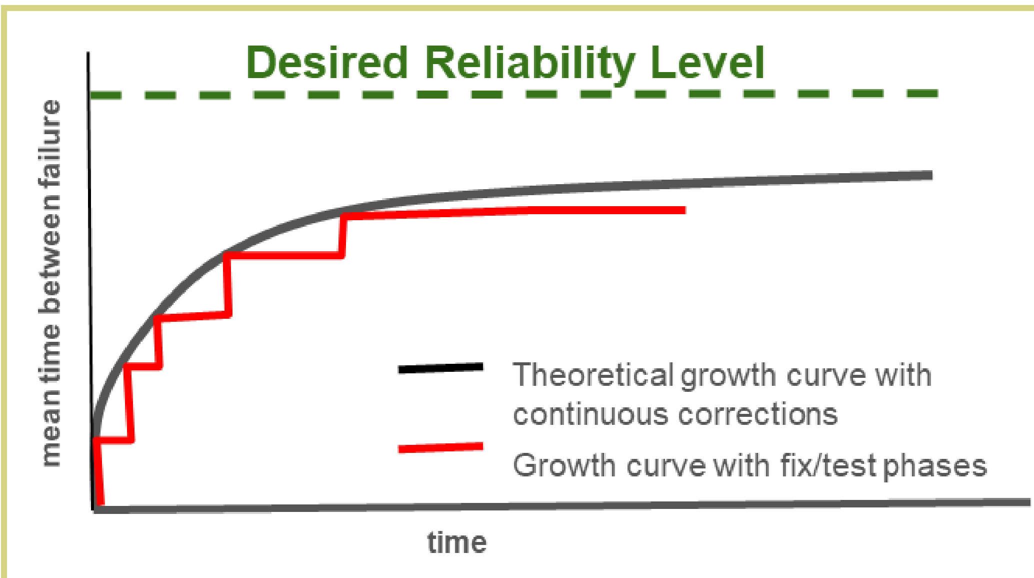

Figure 7: Growth Curve with Correction Phases and Poor Performance.

7.3 Failure Mode and Effects Analysis (FMEA)

Failure Mode and Effects Analysis (FMEA) is a systematic reliability engineering methodology widely employed across industries, including aerospace, automotive, medical devices, and increasingly in software and AI system development. FMEA provides a structured approach for identifying potential failure modes within a system or process, analyzing their effects on system performance, and assessing the risk associated with each failure mode. The analysis typically involves cross-functional teams that

[18] Note behavior where reliability improves in discrete steps can be seen in complex military weapon systems too. This is because weapon system development often has a ""test-fix-test"" structure with distinct test and corrective action phases.

systematically examine each component or process step to determine how it might fail, the consequences of that failure, and the likelihood of occurrence and detection.

Each failure mode is typically scored using three aspects of a failure, severity, frequency, and detectability (probability that the event would *not* be detected before the user was aware of it), which are combined into a Risk Priority Number (RPN) to guide corrective action prioritization. For complex agentic AI systems, FMEA becomes particularly valuable as it can help identify failure modes that emerge from agent interactions, user interactions, or unexpected behaviors that might not be apparent when examining individual agents in isolation. The systematic nature of FMEA aligns well with the iterative development and continuous learning characteristics of AI systems, providing a framework for capturing and addressing failure modes as they are discovered during system operation

About Reins AI

Reins AI assesses Generative AI applications to verify and guide them toward efficiency, quality, and suitability standard compliance. Our services include quality assessments, verification designs, and improvement guidance, with expertise in rigorous market validation, product design, and quantitative evaluations for building products that make meaningful impacts in the work of human experts.

Marisa Ferrara Boston, PhD
Managing Partner, Reins AI

Reins AI was founded in 2023 by Marisa Ferrara Boston. Marisa is an expert in designing and evaluating technology that augments the most human aspects of our work: collaboration, organization, and the transmission of knowledge. With successes in industries spanning financial audit, customer service, crowdsourcing, R&D, and healthcare, she understands how years of productivity-oriented augmentation have revolutionized the speed of business at the cost of maintaining and enhancing organizational knowledge. She has held roles in big tech and consulting, where she was a hands-on scientist, builder, and manager. She holds a PhD in Cognitive Science (double major, Computer Science and Linguistics) from Cornell University, where she focused on applying information-theoretic measures to human cognitive models.

Heather Frase, PhD
Evaluation Lead, Reins AI

Heather Frase, PhD is the CEO of Veraitech and Senior Advisor for Testing & Evaluation of AI at Virginia Tech's National Security Institute. Her diverse career spans roles in defense, intelligence, policy, and financial crime. Her current work focuses on developing and supporting the evaluation of AI systems, improving reliability, and aligning performance with real-world use. She also serves on the OECD's Network of Experts on AI and on the board of the Responsible AI Collaborative, which researches and documents AI incidents.

Effi Georgala, PhD
Agent Reliability Lead, Reins AI

Effi Georgala, PhD is an impact-focused AI leader who designs and delivers reliable, human-centered systems at the intersection of research and production. She holds a PhD in Linguistics & Cognitive Science from Cornell and brings 10+ years of experience at Microsoft and Nuance, where she led end-to-end AI initiatives across enterprise and healthcare – from early feasibility through validation, and iteration. Effi specializes in the reliability and repair of AI systems, turning monitoring into reproducible fixes and measurable reliability gains.