Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

When Systems Are Uncertain: Human Reliability Engineering in AI-Driven Production

Sreejith Kaimal
Principal SRE, C3.AI, USA

ARTICLE INFO

Received: 12 Feb 2026

Revised: 15 Feb 2026

ABSTRACT

Existing work on reliability engineering is generally based on a deterministic model of the world, and interpretable modes of failure. Such assumptions are not valid for production AI systems, motivating the discussion of the model change from machine-centric to human-centric reliability constraints. The latter are defined by human operators' cognitive load, trust calibration, and interpretation of probabilistic signals. Challenges include cognitive overload from non-deterministic failure diagnosis, trust in automated incident response, alert design for probabilistic systems, investigation of opaque failures, and maintaining human agency in automation. These challenges illustrate that the reliability of AI systems requires a focus on human-machine cognitive collaboration, rather than simply a focus on performance. In this context, observability, alerting and automation can be reengineered to the cognitive limits of humans and to prevent skill loss. From a sociotechnical perspective, reliability can be understood as an emergent property of human-AI interaction. This provides the new basis for managing uncertainty in production environments beyond the limits of deterministic models.

Keywords: Human Reliability Engineering, AI Operations, Cognitive Load, Trust Calibration, Probabilistic Alerting

Introduction

Classic SRE is grounded in deterministic failure signals and interpreted system behavior. An infrastructure component either behaves in a normal manner, or it fails in a mode whose failures produce an alert with a well understood remediation. These assumptions are violated for AI production systems because their failure modes are stochastic, system behavior is nondeterministic, and decision boundaries are hard to reason about. The non-deterministic failure modes of machine learning systems are different from those of conventional software systems [1]. Reasons for this group of problems are the difficulty of identifying data requirements, the quality and quantity of data, and the differences between data science and software engineering work. Rather than machine reliability, it needs to be thought about the perceptual reliability of the operator's mental model, trust calibration, and alert interpretation. The goal is to build AI as critical infrastructure, where researchers simply need to monitor how CPU usage affects human attention, understanding, trust, and decision making under uncertainty. Successful AI-based production systems are not only characterized by their model accuracy or system uptime, but also depend on how well this human-machine interaction is managed, enabling operators to diagnose, intervene, and trust in fundamentally uncertain environments.

A Unified Human-AI Reliability Framework (H-AIR)

Human-AI reliability is the sustained ability of a sociotechnical system to achieve its intended outcomes in conditions of uncertainty by jointly optimizing all four pillars (see below) without exceeding user cognitive limits or diminishing human agency and accountability over time and in interaction with particular AI systems.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Cognitive Dimensions (Human Limits as First-Class Constraints)

Humans are bounded, contextual reasoners, not infinite exception handlers, and hence AI systems must recognize and exploit the bounds of human attention, working memory, stress-induced performance decrement, and skill decay under automation. This is particularly problematic in Security Operations Centers (SOCs) where alert pipelines are often 80% false positives due to poorly calibrated detection rules [15]. Alert noise overloads analysts, delays threat response, and eventually tires them out. Analysts need to separate signal from noise. Compounding the problem is 'automation bias', where analysts over-rely on AI recommendations and fail to think critically in high-priority cases [15].

The first principle states that the overall system complexity needs to be within human cognitive bandwidth, without any open-ended cognitive adaptation on the part of the human users [15]. This entails a number of requirements. First, there are cognitive limits on operators, namely that operators should be limited to three or fewer concurrent cognitive threads, as anything more than three degrades their ability to make high-quality decisions. Second, systems should utilize trend rather than instance-level cognition. Third, interfaces should make clear why now before what to do, to help operators prioritize their attention. Fourth, conditions for making sense of a critical incident should be designed that slow, deliberative thinking is favored over fast, intuitive pattern matching, even when the decision is time critical (e.g. sonar operators).

Information must be structured in terms of cognitive load, relevance, and urgency, and within the limitations of human operators [15]. Avoid "monitor-only" roles for extended periods. An interesting challenge for the designers of computational systems is that humans are only able to attend to information for limited time periods in non-active roles - which is called vigilance failure. Systems should allow for metacognition by including user limits. For example, control relinquishment via confidence should be followed by the AI ceding control to humans to prevent performance loss in demanding operational environments [15]. For example, dashboards may be organized around questions, such as "Is the model drifting?" instead of displaying forty metrics. In progressive disclosure architectures, levels of information are presented in the following order: summaries, drill-downs, and evidence. Systems should incorporate decision pacing mechanisms for high-risk actions, to implement a structured delay and reduce the chance of impulsive actions in time-critical, high-stress environments.

Trust Calibration Mechanisms

Trust calibration results from the perception of the system's competence, not its intelligence, as both automation bias (over-trust) and lack of trust (i.e. rejection of valid automation) diminish the performance. AI trust is a joint measure of competence and intention [16]. Humans anthropomorphize machines as social agents. This understanding of trust building would seem to imply some sociocognitive understanding [16]. Because of this tendency, trust calibration is not something that can simply emerge when social agency is given to an AI system.

The effects of transparency on trust are somewhat mixed. Some studies find that transparency does promote trust when the information is visually or textually presented. Others find that more transparency can reduce trust for some tasks [16]. Although only 19-21% of survey respondents believe existing AI safety measures are sufficient, safety was one of the most important factors on trust calibration. This indicates that more direct trust calibration interventions are needed beyond improving objective levels of safety [16].

Trust calibration typically must use several levers. Confidence scores can be based on prior accuracy instead of model logits, providing users with empirically-grounded estimates of reliability. The failure modes should be made explicit, and it should be obvious when the system is uncertain about its output, and this information should be made clearer when the system's confidence is low. Periodic trust resets after model updates can prevent outdated mental models from interfering when system

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

capabilities change, and operators should be trained to identify model strengths and failure modes. This can create an accurate and balanced mental model.

Well-designed post error explainer approaches can partially restore trust, although effectiveness depends on several factors [16]. Trust can be developed in a focused manner via feedback loops and approaches that promote transparency such as explainability panels and uncertainty scoring. This enables on the fly scalable autonomization while maintaining governance, accountability and operation safety [15]. This gradual increase in trust can be modeled as: **Trust = f(past performance, explainability, reversibility)**.

This formalization leads to the following design pattern: irreversible actions, where errors cannot be undone, should require human confirmation. Situations with ambiguity should use counterfactual presentations, for example by asking the system to list which decisions were considered and rejected. Repetitive actions that are known to be reliable can be automated, since an error could be corrected in the next repetition.

Interpretability & Observability

Interpretability requires, in addition to post-hoc audits, reasoning about the current operator's question: Why did the AI decide to act? What would have changed? Has this happened before? Is this within normal operating parameters? However, many machine learning algorithms such as neural networks lack intrinsic human interpretability [16]. Deep learning models such as long short-term memory networks and transformers are increasingly being used for security applications, although there is a lack of explainability for such models [15][16]. However, explainability reduces perceived risk associated with AI [7]. Infrastructure that eases real-time observability supports situational awareness when humans and AI jointly detect, make sense of, and predict future threat trajectories. The AI should make its reasoning and uncertainties visible in a maximum value to the human analyst [15]. Interpretability techniques should support teammate awareness by making the role, cognitive state and task load of agents in a human-AI team known to other teammates in the collaborative team. Second, world awareness provides a simultaneous, continuous view of the operating environment, including threat regions, the state of the system, organizational goals, and countermeasure efforts [15].

A trade-off between explanation richness and cognitive feasibility may warrant going beyond academic interpretability methods to utilitarian explanation methods that are explicitly designed to enable decisions [16]. Similar to this, case-based reasoning uses past similar cases, relying on the human ability at analogical reasoning. Counterfactuals describe 'what if' scenarios to describe how the recommendation would change based on a change to the model input; e.g. "If feature X were 20% lower, the model would have recommended the user suggest action Y to the online queue". Directionality allows features to specify how they impact the recommendation, rather than relying on model operators to interpret SHAP plots or the like. Behavior fingerprints allow model operators to compare the model behavior to historical behavior fingerprints for similarity, to check for important regression. In addition, model lineage linked to downstream effects tells users which model version produced which outcomes, creating accountability and imparting confidence in what the system can do and that designers are addressing systemic sources of bias. This helps all user groups understand the system better and reduce bias [16].

The worst anti-pattern is to offer some sort of post-hoc interpretability that is not real-time, and is solely useful for an audit. The interpretability must be operational, meaning it must happen in real-time at the decision point, rather than after the consequence of the decision has already manifested itself.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Automation Governance

Automation governance focuses on accountable ownership roles, which can be distilled to a guiding principle: autonomous execution should not be possible without a named accountable human role. This principle rests on the idea that governance is about owning decisions and delegating agency and responsibility to people and AI systems and not about compliance artifacts. Trust and capability of human-AI systems increases when human decision making is at the forefront and when systems are conditionally or fully autonomous with humans performing oversight and planning [15]. From this spectrum, a task allocation principle can be inferred: routine, plain tasks should be delegated to agents with low complexity, while high level decisions should remain with human analysts. This has the effect of improving MTTR, maintaining situational awareness and trust in decision authority, and preserving accountability to a human [15].

Human-in-the-Loop (HITL) design patterns systematize when to involve humans and when to fully automate, based on the argument that humans should intervene only in case of uncertainty, value judgment or irreversible consequences, rather than being a back-up in case something goes wrong or for error resolution. Common models use approval gates, where the AI proposes a solution that must be signed off by humans. For example, AI-based medical diagnosis or financial trading may have socially important consequences that require human accountability. With exception handling, AI operates autonomously until confidence or policy thresholds are violated, prompting its decisions to be escalated to humans. This supervisory control positions humans to monitor trends rather than individual actions, reducing alert fatigue [21]. Human-as-teacher patterns commonly use feedback loops like RLHF and active learning to correct models. Kill switch authority grants humans the ability to shut down models in the event of catastrophic or ethical failures. The anti-pattern is "human as error handler": full human participation at machine speed with little domain context. The rule of thumb: automate execution and keep the human in the loop for judgment, escalation, and accountability.

Practical governance will set automation boundaries on the types of decisions AI systems can take independently and those requiring human operation. These boundaries might be defined by drivers including the uncertainty of the AI, potential impact on the domain of interest, and the deviation of the current situation from normal operations. Human veto paths should make sure that humans are not slowed down or punished for overriding an AI decision. Lastly, when humans are auditing, they should do so on the decision, not the prediction, as this will make them accountable for the consequences. To allow for continual reassessment and reinvention, autonomous behaviors should include sunset clauses. Furthermore, responsibilities should aim to build on the strengths of both humans and AI in complementary ways [15]. It may also be that some tasks are more suited to AI models, such as the processing of large datasets, predictive analysis of patterns and autonomous decision-making on well-structured tasks, whereas contextual reasoning that accounts for tacit knowledge, understanding of business processes and organization, ethical reasoning and competing values, would be more human domains. These complementary capabilities suggest AI should mainly handle computations, with humans performing normative reasoning as contextual experts in decision making.

The legal and regulatory dimension to trust is part of the governance element. The General Data Protection Regulation (GDPR) that became effective in 2018 by the European Union is the first legal framework for high privacy, algorithmic transparency and accountability [16]. These rules and regulations create the baseline for AI governance, specifying not only organizational policies but also legal obligations. For higher levels of autonomy, the whole SOC process pipeline is performed by the machine, without human oversight, including threat intelligence data ingestion, threat hunting, auto-containment and detection model updates, among others. Only a human can govern, audit and manage the policies implemented within an organization [15]. Humans are responsible for the continuous review and assessment of AI decisions and are expected to provide overall calculated insights, but are removed from day-to-day operational work. They also need to address issues such as

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

data privacy, accountability, privacy regulation and perceived uncertainty around the adoption of AI [15, 16]. This division between operational tasks and human judgment allows the organization to prioritize the improvement of operational capabilities over the preservation of judgment necessary for organization-wide calculated and normative objectives.

Cognitive Load and Mean-Time-to-Understanding

The Non-Determinism Problem

While classic system failures are binary and produce yes/no errors (server down, disk space full, network latency too high), AI system failures are non-deterministic and incremental (performance drift, distribution shift, unexpected behavior) and thus do not typically trigger availability alerts. Research in the deployment of machine learning has shown that drift of data, models and infrastructure affects many (perhaps most) production ML systems, and may be gradual rather than sudden, leading to the need for active monitoring and adaptation rather than a one-off fix [1]. This results in probabilistic failure narratives that must be inferred from partial and noisy observability, in contrast to deterministic failure modes characteristic of infrastructure problems.

Cognitive load may be further increased by the challenge of mapping a single error in the recommendation system to multiple sources of failure, such as training-serving skew, concept drift, and performance degradation, which interact with each other in different ways, and carry implications for various business metrics. The ACM survey revealed that people in the field encounter interdependent, interacting failure modes throughout the ML lifecycle, indicating that they should be testing multiple hypotheses at once rather than following debug paths linearly [3]. This is in sharp contrast to many operational scenarios where the root cause of an error can be tracked down easily.

Signal Complexity and Interpretability

The core of this design is the Conv layer which learns feature maps with respect to spatial or structural characteristics by using shared-weight filters. These filters, in conventional deep learning approaches, tend to be opaque, providing a high predictive capability with reduced interpretability. Understanding how specific activations lead to downstream predictions can be challenging. Figure 1 provides a visual overview of the interpretable Conv-layer is proposed. On top of the convolution operation, interpretable Conv-layer adds masking and filter-wise loss attribution, that is, for each filter, a masked area of the input and a contribution to the output loss are clearly defined. This enables comprehension of the model's behavior at the level of individual features.

Conversely, AI-specific observability signals, such as a model's confidence scores, feature attribution scores, or embedding drift statistics, only provide a partial view of what happens in a model. Furthermore, for machine learning observability, interpreting machine learning signals often requires advanced knowledge of the domain. As a result, interpretable deep learning models mainly seek to construct intermediate representations of the neural network itself that can explain how learned features contribute to the output. Loss curves mainly convey global performance on validation/test sets, but do not explicitly provide visibility into the operator's model. Operators may misinterpret the signals from AI if the model is not represented in an interpretable manner on the sub-component level (e.g., filter-wise loss attribution in convolutional layers). This may entail a higher cognitive load on operators and longer time-to-diagnosis between the model's behavior and system health. This architectural integration of interpretability within Conv-layers provides a principled approach to bridging the model activation outputs with more intuitive human interpretations of the model activity.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

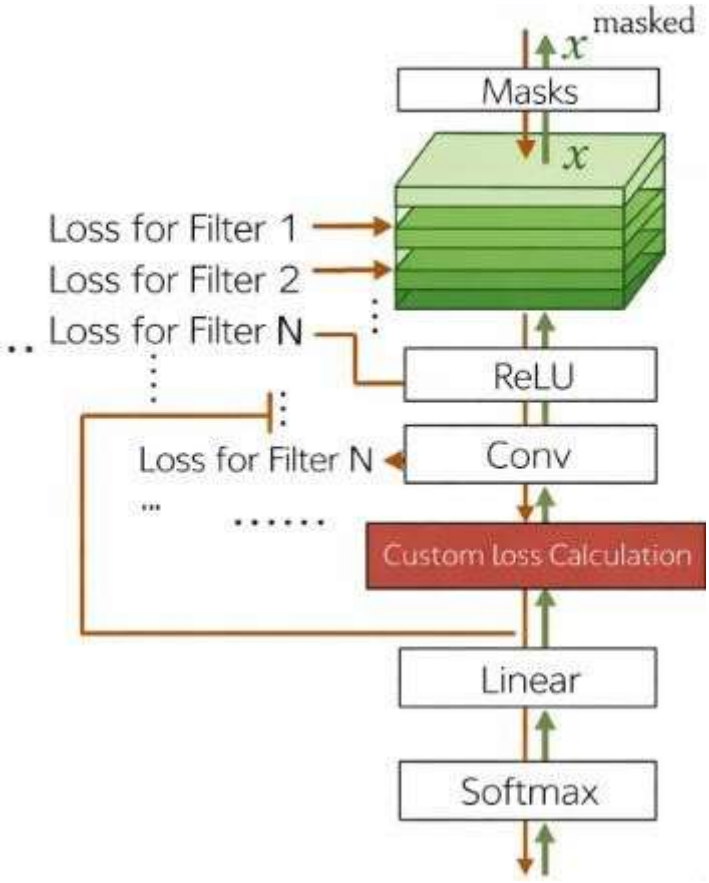

Figure 1: Structures of an ordinary conv-layer and an interpretable conv-layer [2]

Designing for Cognitive Efficiency

To help reduce mean-time-to-understanding, careful design of diagnostic information flows is needed. Monitoring and observability tools were one of the most-cited operational bottlenecks in ML deployment workflows in the ACM survey, with practitioners citing a need to tie model health and system behavior together in a unified view [3][turn1search4]. Integrating a structured view of business metrics, an intermediate view of ML health metrics and a low level view of system telemetry allows operators to construct a mental model of the system behavior more quickly than using a separate dashboard for each metric. Visualizations of distributional shifts, uncertainty bands, and trends allow for gradual performance degradation detection without meaningful statistical analysis [2].

Further down the stack, probabilistic system visualizations should use confidence intervals, distribution shifts, and time-series trends to expose gradual degradation rather than just point estimates often used in dashboards. Comparative views, showing current behavior in relation to historical baselines, help operators quickly identify anomalous patterns without deep statistical reasoning for every incident [2].

Non-technical measures, however, are more tied to cognitive efficiency metrics seeking to convey human-AI collaboration quality, such as Operator Understanding Velocity (OUV), defined as the speed at which an operator correctly gets an understanding of an AI's behavior, and assessed as time to correct explanation in simulation, accuracy of prediction of next action, and error rate reduction over historical repetitions. Informed by Situation Awareness (Endsley) and Mental Model Theory (Johnson-Laird), the OUV is a measure of whether operators are forming and maintaining accurate models of the state and behavior of the system [17]. A high OUV indicates operators are rapidly

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

updating their mental models of the AI after behavior change due to a model update, distribution shift, or unknown operational context. This leads to incidents being resolved quickly and reliably.

Trust Calibration in Human-AI Incident Response

The Over-Trust Failure Mode

Automated diagnosis and remediation systems promise to decrease the workload of system operators and reduce the time it takes to resolve large numbers of problems. However, these systems also run the risk of automation bias, a reliance on automation that takes precedence over human reasoning. For example, in decision support studies where clinicians view CDSS output, automation cues can increase diagnosis accuracy from 50.38% to 58.27%, but also cause more clinicians to move from the correct to incorrect decision in 5.2% of cases [5]. It shows how the repeated success of the system causes operators to pass responsibility to automation.

Human-factors literature has studied automation misuse and complacency as single points of failure. Automation misuse occurs when operators take the automation output as infallible. Automation complacency occurs when operators do not sufficiently evaluate the automation output. Both occur regardless of automation reliability and operator expertise [6]. Such over-trust is relevant in an AI-augmented incident response because the operators might accept and implement automated root cause and remediation, including for edge-case incidents.

The Under-Trust Barrier

On the other hand, operators may under-trust or disbelieve useful automation recommendations as a response to prior automation errors or false alarms, resulting in slower responsiveness and increased diagnostic demands. Human-automation trust models indicate that operator trust may be a dynamic factor and that the trust increment is smaller than the trust decrement (the trust decrement is a larger effect than the trust increment) [7]. Failing to reject accurate automation suggestions can lead to later responses, increased mean-time-to-understanding, and ultimately failing to act in situations where a fast response is expected.

Restoring Calibrated Trust

First, transparent trust requires that systems convey their capabilities and limitations and automated diagnostic tools communicate their uncertainty bounds, their known failure modes and supporting evidence for their conclusions. Core research in the trust in automation literature shows that for automation to be effectively used, operators need to have an understanding of automation reliability and conditions of success and failure [7]. As such, AI-based incident response systems should focus on hypothesis generation and speeding up human decision-making, rather than diagnosis.

Trust State	Automation Acceptance Pattern	Override Behavior	Recovery Efficiency	Escalation Tendency
Over-Trust	High acceptance of incorrect recommendations	Minimal appropriate overrides	Moderate efficiency	Low escalation
Calibrated Trust	High acceptance of correct recommendations	Balanced override decisions	Optimal efficiency	Appropriate escalation
Under-Trust	Low acceptance of correct recommendations	Excessive inappropriate overrides	Poor efficiency	High escalation
Post-False-Positive	Very low acceptance	Persistent excessive overrides	Severely degraded efficiency	Very high escalation

Table 1: Trust Calibration Outcomes in AI-Assisted Incident Response [7]

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Developers should not present AI outputs as a final diagnosis, but rather as hypotheses informed by uncertainty with a rationale, which could reduce the blind acceptance or rejection of automation. Such a presentation encourages operators to combine automation outputs with their expertise, as well as their knowledge about the system and about its context [7].

Trust calibration can be assessed by the concordance of operator trust and system performance. This is known as Trust Calibration Accuracy (TCA). TCA can be calculated by the correlation coefficient between the operator trust ratings and the system's accuracy, the percentage of over-trust when confidence is low and the percentage of under-trust when confidence is high [18]. TCA is grounded in the Human-Automation Trust Models proposed by Lee and See [19], positing that the best performance is given when trust has the right degree of calibration to automation reliability. TCA can be operationalized through periodically asking operators to assess the trustworthiness of a limited number of AI recommendations and comparing these to the outcomes of the selected recommendations. For perfectly calibrated AI recommendations, operator trust correlates with empirical accuracy; for miscalibrated recommendations, it tends to diverge in one systematic direction.

Alert Fatigue and Probabilistic Signal Design

Entropy in AI System Alerting

Customary alerting relies on clean violations (e.g. a metric exceeds a static threshold). In contrast, AI systems produce non-binary probabilistic alerts (e.g. confidence drift or distribution shifts) with soft, uncertain boundaries that yield large numbers of uncertain alerts. High alert burden of different types is a leading contributor to alert fatigue, a human behavioral phenomenon in which operators override or disable alarms due to excessive irrelevant alarms [8].

One of the best studied areas of clinical systems that include alerting are CDSS. Evidence that alert fatigue occurs in clinical practice using CDSS has been reported by various systematic reviews. Override rates for diagnostic alerts were found to be as low as 46.2% and as high as 96.2% depending on implementation and alert type [9]. Overloading operators with a collection of non-actionable alerts may desensitize the operator to alerts that indicate genuine problems.

This can result in alert storms due to the fixed threshold, making small fluctuations in model confidence or other internal metrics trigger alerts, and training operators to ignore alerts as noise rather than actionable issues.

Confidence Bands and Contextual Thresholding

Human factors research has shown that the relevance and context of alerts are the best predictors of operator engagement with the alerts. For example, CDSS drug interaction alerts with low specificity (generic alerts) are overridden at very high rates due to lack of clinical relevance [10]. In general, alert systems can incorporate confidence bands and context-based thresholds (for example, alerts be adjusted based on times of day or anticipated workload in which abnormalities would be expected) to improve signal fidelity and reduce the generation of non-useful alerts.

If alerts are configured to be within the expected distribution and operational context, it becomes easier to catch these changes in behavior. Because AI and ML systems intrinsically provide probabilistic and variable output.

Probabilistic Alert Framing

Tiered and multi-resolution alerting place alerts (and thus human attention) into levels based on confidence level and the impact of the situation. Human-factors reviews have also called for limiting low-priority alerts that do not merit immediate attention. This in turn helps cognitive processing of relevant signals [8]. Probabilistic alert framing provides the operator additional information about their decision by presenting an event's probability and uncertainty, allowing for understanding the trade-off of slight metric drift against operation risk. Probabilistic signal design, in which monitoring

systems communicate the statistical distributions and confidence intervals of their measurements is related to statistical process control. For example, reporting alarms in terms of statistical importance and expected behaviors instead of triggering arbitrary thresholds can reduce noise, ease mental load, and allow operators to reason about uncertainty in ways that align with their goals [8].

When evaluating the operator performance using alert load, cognitive load should be measured over time intervals relevant to the operational context. SCLI is an estimate of the cognitive load that can be sustained without performance degradation. While it may be necessary to report peak performance measures, the more relevant measure is that of sustained performance over the duty period, or an on-call duty period. Measuring tools include the NASA Task Load Index (NASA-TLX), error rate over time, reaction time drift, as a precursor to fatigue, and the percentage of missed anomalies over time. Building upon Cognitive Load Theory, and the literature on the vigilance decrement, SCLI is characterized as a phenomenon whereby sustained effort degrades human performance in predictable ways [20]. SCLI can be measured or estimated by collecting baseline performance metrics in the first hour of an operator shift, and the degree of performance degradation within the shift. Sustainable cognitive load occurs when performance metric does not change during the operational period, while unsustainable cognitive load occurs when performance metric decreases during the operational period, regardless of subjective engagement levels.

Operational Playbook: Designing Probabilistic Alerts

Most relevantly, such a shift from threshold-based alerts to probabilistic alerts places the alerting focus on where humans are likely to be most useful with their decision-making rather than on threshold values themselves, thus providing better focus for decision-centric alerts to remain both cognitively tractable and useful during the operations of interest. First is decision-relevant alerts, current alerts are raised when an anomaly crosses a threshold, e.g. "confidence dropped below 0.85", even though the anomaly may not have been a net detriment to decision-making. Decision-relevant alerts answer the question "Does action by the operator now prevent a worse outcome than waiting?" To answer this, alert logic must also include contingencies for how the operator may act and the results of those actions. In this sense, decision-relevant alerting is moving up the pyramid so that the alert is providing decision support rather than simply detection.

The second design principle is alerts by decision context. Cognitive fragmentation is caused when an operator must decide how signals relate to each other and to the operational conditions. Contextual bundling can be further extended to the situation when the alerts are for the same decision context and can be bundled into the same frame for the decision [8]: instead of alerting "model confidence decrease", "input distribution shift", and "downstream metric degradation", the alert engine could alert "Payment fraud detection showing correlated degradation signals across confidence, inputs, and business metrics - typical pattern preceding false negative surge". This however reduces the cognitive load by presenting data at the level of a decision instead.

The third principle states that false negatives and false positives might not be equally costly. In practice the costs of errors are not necessarily equal [9]. False negatives, failing to detect a disease, have a very different implication from the costs of a false positive, using the test when it is not needed. This occurs, for example, in fraud detection, where the cost of failure to detect fraud is very different from the cost of misclassifying a legitimate transaction [8]. Alerts should expose the asymmetric costs of taking an action versus not taking it, so operators can make the best decision under uncertainty, and the cost of an alert can be more or less aggressive depending on the business risk.

Fourth, the expected value of the intervention based on the prior estimates of the intervention's effectiveness. A probabilistic alert with a sense of urgency provides perception into the rationale, allowing the operator to prioritize competing demands on their attention based on metric, rather than intuition [8]: "Intervening now prevents $12K average loss in 72% of similar cases." Work: "find and manually review transactions matching the chargeback pattern from Q3 2024 before the next batch is processed." It does this by estimating the expected value (12k*72% = $8.6k expected savings), providing context of a similar chargeback pattern from Q3 2024, associating the behavior with a

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

business outcome (avoiding chargebacks), explicitly specifying what item of work to do (find and review the transactions), and stating a deadline (before the next batch is processed).

Cognition is also taxed with non-decision supportive alerts, which usually need to be aligned with decision-oriented alerting principles before they can be properly implemented in practice. Alerts should be triggered when an operator can affect the probability of that outcome. They should not be triggered when some statistical threshold is not reached. Second, the recommendation to the operator should be based on what succeeded in the past. Third, any technical signal should be translated into business terms. Fourth, the alert should specify what actions should be taken, not simply what the problem is. Fifth, the time context should identify why to act, not why not to act, by providing time sensitive opportunities or threats. Sixth, the cost of acting and the cost of inaction should be placed in context for operators. Finally, alerts should include a reasonable and timely means for operators to defer or dismiss them, as operator judgment may be more accurate than algorithmic judgment of the criticality of the alert.

One way to avoid this is to shift away from anomaly detection to decision support, using alerts as a collaborative interface to supplement human judgement. Alerts should be defined around what the operator needs to make good decisions with the system, rather than around the technical properties of the system behavior. This can help operators be more effective and help calibrate AI models better in production.

Incident Investigation in Opaque Systems

The Explainability Gap

Incidents generally follow a logical sequence: initial deployment causes bug, configuration change causes connection pooling problems, infrastructure failure causes other services to break. AI system incidents do not follow this more straightforward pattern. The behavior of models, controlled by millions of learned parameters interacting with production data, is not easy to interpret. An operator cannot examine code branches to identify why recommendations suddenly favored certain categories; the decision boundary is implicitly encoded in the model weights and is a function of the training data and optimization dynamics.

This opacity extends to training data defects such as label noise and sampling bias, which may only change model behavior in subtle ways. In one ML incident investigation study, data defects were found to take much longer to uncover than code defects [11]. Data errors differ from code errors in that debugging is not as simple as putting breakpoints and printing stack traces. They require large amounts of data and their statistics to be analyzed. Investigative work involves different mental skills and tools from software debugging.

A lack of tooling and platform support makes it difficult to debug failures and faults in production AI systems. There is currently no equivalent of logs, metrics and traces that are familiar to developers in software systems, which can make it difficult for operators to monitor production AI systems in a similar way. Most interpretability tools produce fragile, post-hoc explanations. An operational ML stack has limited debugging support. Tools for root-cause analysis require manual tracing of links through data, models and infrastructure to isolate ML failures from other production issues. Existing drift detection methods for data drift, concept drift and label drift rarely allow timely intervention. Additionally, accountability in practice is difficult because it is technically difficult to trace the source of negative consequences to either a version of the model or data slices. Most AI failures are not sudden catastrophic failures, but rather slow degradation over time. New tools for feature attribution dashboards, counterfactuals, lineage and data versioning, and clustering semantically similar errors could also deliver strong value, but are not yet sufficiently mature for high quality integration into production AI systems [11].

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Building Investigative Frameworks

Structured hypothesis generation is required to enable efficient incident investigation. Specifically, operators should ask: how is the failure space of the selected model structured? Examining if any changes to the input distribution, the model, the downstream usage, or the operational environment may have caused the incident. Successful ML incident investigations tend to follow a structured diagnostic framework, with teams who follow such frameworks able to effectively and quickly resolve incidents and identify their root cause [11].

For comparing a model against a baseline, graphing its predictions against fixed historical checkpoints allows for immediate visualization of how its behavior evolved. To find covariate shift, graphing the present input data distribution against the training data can save researchers from needing to correlate them, leaving their cognitive resources free for interpreting the changes.

Counterfactual analysis is another method, whereby inputs are manipulated, and the corresponding output observed to gain perception into the model's decision boundaries. Identifying and removing non-contributing features can also be helpful by reducing the workload of examining many features not of interest and focusing on features of interest in the system. Tools for automated counterfactual analysis considerably reduce the effort required on a production ML debugging task [12].

Documentation and Knowledge Transfer

The takeaways from AI incidents are not easily distilled into runbooks, as these are context-dependent (e.g. the current data distributions, what model versions are currently deployed, the current business logic) and very quickly obsolete. Runbooks must not only contain steps to remediate incidents, but also document heuristics for assessing what to do in other types of incidents.

Postmortem documentation of failure modes should be specific to the event. For instance, instead of documenting "model confidence dropped due to seasonal data shift", it can be more useful to document "monitor for correlation between external events and model performance metrics when input distributions differ from the training set." Documentation of ML incidents with a focus on the underlying patterns of failure resulted in better knowledge transfer as measured by the time-to-resolution of future similar incidents than documentation focused on the details of the incident itself [12].

The organization within production teams can impact human-AI interaction. For example, one anti-pattern is siloed ownership of ML and SRE (site reliability engineer) teams over models and infrastructure, and the blame of domain experts when AI outcomes are not as expected. This creates a failure of accountability and understanding at three levels: ML engineers do not see operational failure modes, SRE teams do not understand model behavior, and domain experts do not have direct control over systems producing outcomes for which they are responsible. A better model implements triad ownership, distributing complementary responsibilities across three roles with shared accountability [22]. The ML engineer is responsible for model performance and known failure modes, the SRE is responsible for infrastructure reliability and observability tooling, the domain expert is responsible for outcome evaluation and whether model outputs are consistent with the business problem, and all three roles share an on-call rotation. This creates symmetry of learning about the challenges faced by each other because they do the same work themselves [12].

Cross-domain lessons from aviation, nuclear energy, and medicine suggest developing an understanding of when and why operator errors occur, and creating AI systems that are sufficiently explainable to avoid mode confusion, where an operator can't determine the current mode of operation. In aviation, the use of checklists and standard operating procedures have been determined to be more effective than relying on individual human judgment under stressful conditions, so AI systems should make their current state, intent, and limitations apparent to an operator. Nuclear power engineering assumes that humans will be fallible under stress, and thus applies fail safe, redundancy, procedural discipline, thick walls, and other strategies rather than "smart" interfaces to ensure safety. This suggests that AI teams should design conservatively and expect that human

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

operators will not be optimal. In healthcare, decision support systems should be optimal when they are merely advisors. Too much automation breeds automation bias and decreases vigilance. The role of AI in the workplace should be to augment cognition, and allow decision-makers to be involved with decisions that matter rather than be vigilantly observing a system.

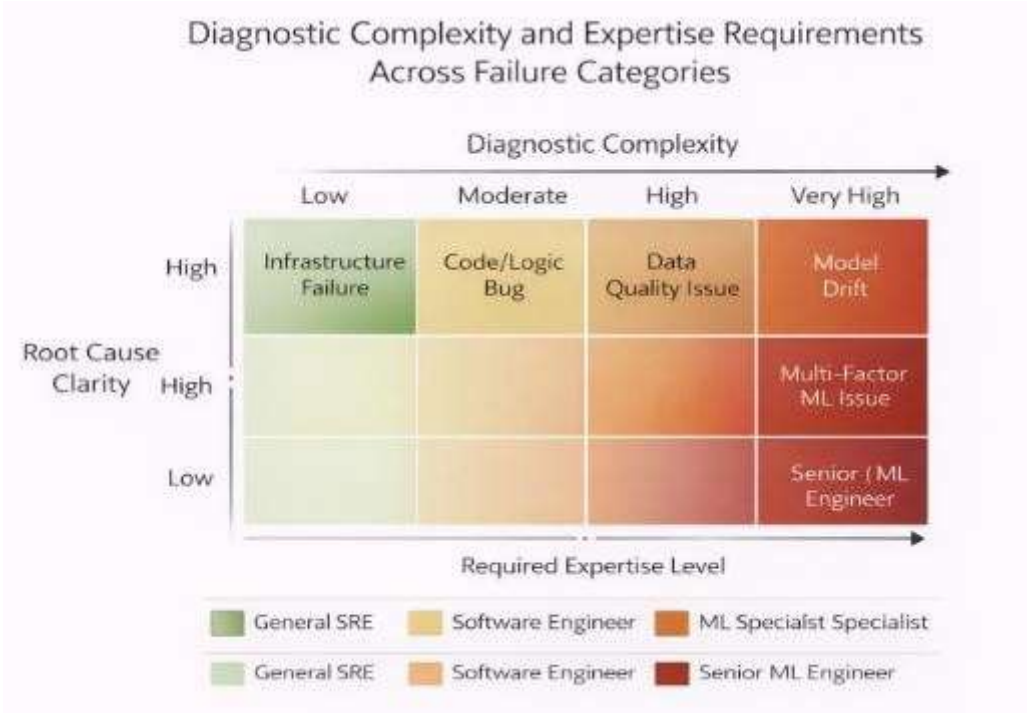

Figure 2: Diagnostic Complexity and Expertise Requirements Across Failure Categories [11, 12]

Automation, Orchestration, and Human Override

The Remediation Automation Spectrum

Automated remediation of AI systems can be as simple as a rollback mechanism or as complex as self-healing processes. An automated AI system may roll back to a saved model checkpoint upon regression in confidence metrics. More aggressive strategies include simply retraining on the most recent data, dynamic exclusion of drifted inputs, and real-time ensembling. More automation allows for faster response times, but increases the chance of unnecessary model intervention [13].

However, temporary data anomalies could lead to unnecessary rollback, at the potential expense of the modeling. Also, user experience will not be optimal in case of failures, as error detection and recovery may take time. Empirical studies of ML deployment strategies find a tradeoff between speed and accuracy: while automation drastically reduces mean incident duration, a high proportion of automated actions configure systems incorrectly [13]. The tradeoffs are difficult to codify because optimal remediation actions are context-dependent (business criticality, time of day, degradation severity, and diagnostic confidence) and require human judgment [3].

Designing for Human Override

Automation should be transparent. Operators must understand how the automation system reasons to assist or override it when necessary. An auto-remediation system might put forth a message like "Rolling back to checkpoint from earlier today because confidence scores dropped significantly and input distribution shifted considerably from training baseline." Research in explainable AI in operational settings has shown that transparent automation reduces inappropriate human

intervention, because the reasoning of the automation is visible, and humans have override capacity [13].

Override interfaces must provide urgency gradients to their operators, such as immediate kill switches to cancel automation and revert to manual operation in incident situations. In practice, graceful overrides of this kind enable automation to progress through states, including pausing to investigate a trigger or changing the remediation threshold based on the actual operational conditions [5]. This prevents the systems from being in either fully automated control or in full manual control as in brittle automation [4].

Feedback loops between overrides and automation can be used to improve the reliability of a system. For example, if operators often override an automation's decisions, the automation might not be properly calibrated. Analyzing overrides can also help diagnose and identify systematic failures in automated reasoning [14]. Learning from operator overrides reduced false positives of remediation actions throughout the operational lifetime of the automated system [14].

Preserving Operator Agency

Heavy use of automated systems leads to deskilling through disuse atrophy. When a problem is fully solved by automation, the operator might never have a chance to manually solve the problem. This creates a dangerous dependency: the more dependable automation becomes, the less practice humans have solving problems without it [4]. Studies manipulating skill degradation in automated settings show meaningful loss of manual troubleshooting skill when operators have spent long periods of time interacting with automated incident management solutions [14]. Intentional practice for operators, such as periodic manual incident response exercises, is needed to keep troubleshooting skill levels high when systems are stable. Some types of incidents have systems being automated, while others do not. The process of deskilling and rotating the deskilling is a way to preserve the efficiency of an automated system while avoiding the consequences of fully deskilling the system without a human operator [3]. Operator agency is broader than override controls because it can influence the design of automation. Incident response teams are closest to failure modes and edge cases that stress test automation. The open interfaces for operators to share their automation needs and limitations would improve system reliability and engagement with human operators [1]. It accounts for human factors related to reliability in AI systems that are about collaborating with the human operator rather than replacing them [5].

The challenge is this: if the automation is successful, there will be less need for a human to step in when the automation fails (and the organization's experts may erode their experience and ability to handle such incidents). However, as an AI system takes more operational decisions from the human operator, there is reduced opportunity for practice and exposure to edge cases. Countermeasures exist to reduce the decay of expertise while retaining the benefits of automation. This can be achieved through operator training with manual mode drills where decisions are made without relying on automation, like manual flying training even if pilots fly with autopilots. Incentives should be applied to reward early intervention rather than passive monitoring, or rewarding operators who detect and respond to degradation before automation does. Career paths should exist for those who are stewards of systems with deep operational expertise, not just for those who invent new capabilities. Organizations should ensure that budgets for human skill maintenance exist. Expertise has an operating cost and should not be assumed to be obsoleted by automation [11].

Ethical concerns in AI development and production often stem from organizational and cultural issues rather than technical ones. Three general categories of risk are particularly relevant. There are two main scenarios that give rise to moral hazard problems. The first is known as diffused accountability, where multiple parties can impact the behavior of a given AI system but no single party can be directly held accountable for its actions. One is where parties can shift responsibility to each other. The second is where automation is taken as authority, such that organizations can present themselves as not responsible for the human decisions embedded in technical systems. Third, economic pressure can cause the system to be operated without buffers, in ways that compromise its

safety. For example, firms may deploy an AI system without appropriate safety measures or keep a system running in production despite known reliability shortfalls [15]. A key principle is that if humans are to be held accountable for a particular outcome, they must have the necessary authority, knowledge and time. Accountability without authority creates situations in which people are held responsible for actions over which they have no control. Accountability without knowledge creates situations in which people are punished for failures they could not reasonably foresee. Accountability without time creates pressures for hasty decisions that fail to consider all relevant information [21].

Automation Level	Example Remediation Action	Primary Benefit	Key Risk Introduced	Required Human Oversight
Manual	Engineer-driven diagnosis and fix	Maximum control and understanding	Slow response, operator fatigue	Continuous
Assisted	AI-generated diagnostics or suggestions	Faster hypothesis generation	Automation bias, misinterpretation	High
Guardrailed Automation	Automated rollback within predefined constraints	Rapid recovery for known failure modes	Incorrect rollback in edge cases	Moderate
Adaptive Automation	Automated retraining or dynamic feature exclusion	Improved resilience to drift	Compounding errors, unstable feedback loops	High
Autonomous Remediation	Fully automated mitigation without human approval	Minimal downtime	Loss of operator awareness, cascading failures	Critical (post-hoc review)

Table 2: Automation Levels and Risk Trade-offs [13, 14]

Conclusion

Critical infrastructure is moving into a new phase of functional partnership with AI systems, and new measures of reliability beyond availability and latency are needed to track the quality of human-machine collaboration. The proposed metrics of sociotechnical system health, Operator Understanding Velocity, Trust Calibration Accuracy, and Sustainable Cognitive Load Index, are grounded in research and can be operationalized and measured in practice. They provide a measure of the quality of human/AI collaboration that can be used to detect degradation before failure occurs. Structural and cultural measures such as team composition, blameless postmortems for AI systems, and incentives that compensate for human knowledge create a layer that is often overlooked, where technical systems meet human organizations. Ethics recognizes that reliability engineering for AI systems is not value-free when decisions affect human lives. It requires considering accountability, authority, and sociotechnical norms governing sociotechnical systems in which they operate and raise ethical questions within their design. Reliable, safe AI systems can only be built by designing the interactions between human cognitive limitations, machine autonomy, and organizational incentives. People cannot be unsupervised and still build reliable and safe AI systems. The answer is to design AI systems that work within the limits of human attention and capabilities. They make it possible for humans to build and maintain the necessary domain knowledge to deal with the failures of automation. The playbook for probabilistic alerting turns alerting into a decision support tool and helps direct attention where it is most needed.

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Disclaimer:

The views expressed in this article are my own and do not represent the views or positions of C3.AI. I am not representing C3.AI by submitting or publishing the contents in this article.
Sreejith Kaimal (Principal Site Reliability Engineer at C3.AI)

References

[1] Nadia Nahar et al., "Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process," Proceedings of the 44th International Conference on Software Engineering (ICSE), 2022. Available: https://dl.acm.org/doi/10.1145/3510003.3510209
[2] Quanshi Zhang et al., "Interpretable Convolutional Neural Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. Available: https://ieeexplore.ieee.org/document/8579018
[3] Andrei Paleyes et al., "Challenges in Deploying Machine Learning: A Survey of Case Studies," ACM Computing Surveys, 2022. Available: https://dl.acm.org/doi/10.1145/3533378
[4] Raja Parasuraman and Victor Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," Human Factors, 1997. Available: https://journals.sagepub.com/doi/10.1518/001872097778543886
[5] Kate Goddard et al., "Automation bias: empirical results assessing influencing factors," International Journal of Medical Informatics, 2014. Available: https://www.sciencedirect.com/science/article/abs/pii/S1386505614000148
[6] Mary T. Dzindolet et al., "The role of trust in automation reliance," International Journal of Human-Computer Studies, 2003. Available: https://www.sciencedirect.com/science/article/abs/pii/S1071581903000387
[7] John D. Lee and Katrina A. See, "Trust in Automation: Designing for Appropriate Reliance," Human Factors, 2004. Available: https://journals.sagepub.com/doi/10.1518/hfes.46.1.50_30392
[8] Fotios Voutsas et al., "Mitigating Alert Fatigue in Cloud Monitoring Systems: A Machine Learning Perspective," Computer Networks, 2024. Available: https://www.sciencedirect.com/science/article/pii/S138912862400375X
[9] Tahmina Nasrin Poly et al., "Appropriateness of Overridden Alerts in Computerized Physician Order Entry: Systematic Review," JMIR Medical Informatics, 2020. Available: https://medinform.jmir.org/2020/7/e15653
[10] Karen C. Nanji et al., "Overrides of medication-related clinical decision support alerts in outpatients," Journal of the American Medical Informatics Association, 2014. Available: https://pubmed.ncbi.nlm.nih.gov/24166725/
[11] Saleema Amershi et al., "Software Engineering for Machine Learning: A Case Study," Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), 2019. Available: https://ieeexplore.ieee.org/document/8804457
[12] Marco Tulio Ribeiro et al., "Why Should I Trust You? Explaining the Predictions of Any Classifier," Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. Available: https://dl.acm.org/doi/10.1145/2939672.2939778
[13] Dominic Kreuzberger et al., "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," IEEE Access, 2023. Available: https://ieeexplore.ieee.org/document/10081336
[14] Christopher D. and Stephen R. Dixon, "The Benefits of Imperfect Diagnostic Automation: A Synthesis of the Literature," Theoretical Issues in Ergonomics Science, 2015. Available: http://tandfonline.com/doi/abs/10.1080/14639220500370105
[15] Ahmad Mohsin et al., "A Unified Framework for Human AI Collaboration in Security Operations Centers with Trusted Autonomy," arXiv preprint arXiv, 2025. Available: https://arxiv.org/html/2505.23397v2

[16] Yugang Li et al., "Developing trustworthy artificial intelligence: insights from research on interpersonal, human-automation, and human-AI trust," Frontiers in psychology, 2024. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2024.1382693/pdf
[17] Tao Zhang et al., "Using situation awareness measures to characterize mental models in an inductive reasoning task," Theoretical Issues in Ergonomics Science, 2022. Available: https://www.tandfonline.com/doi/abs/10.1080/1463922X.2021.1885083
[18] Kazuo Okamura and Seiji Yamada, "Empirical evaluations of framework for adaptive trust calibration in human-AI cooperation," IEEE Access, 2020. Available: https://ieeexplore.ieee.org/iel7/6287639/6514899/09281021.pdf
[19] John D. Lee and Katrina A. See, "Trust in automation: Designing for appropriate reliance." Human factors, 2004. Available: https://journals.sagepub.com/doi/abs/10.1518/hfes.46.1.50_30392
[20] John Sweller, "Cognitive load theory." In Psychology of learning and motivation, Academic Press, 2011. Available: https://www.emrahakman.com/wp-content/uploads/2024/10/Cognitive-Load-Sweller-2011.pdf
[21] Sushant Kumar et al., "Applications, challenges, and future directions of human-in-the-loop learning," IEEE Access, 2024. Available: https://ieeexplore.ieee.org/iel7/6287639/6514899/10530996.pdf
[22] Alina Mailach and Norbert Siegmund, "Socio-technical anti-patterns in building ML-enabled software: insights from leaders on the forefront," In 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), 2023. Available: https://ieeexplore.ieee.org/abstract/document/10172624/