

Operational Reliability in Healthcare AI

Why reliable Digital Labor requires more than intelligent models

Fabio A. Thiers, MD, PhD

ConverGen AI, New York, NY, USA

ABSTRACT

Complex systems seldom fail because a capable operator made a mistake. They fail because nothing detected the mistake before it propagated. The same structure now governs AI that performs healthcare work, where reliability remains a property of the work system rather than of the model inside it. Long-horizon healthcare workflows are stateful, so an early error can persist even as later reasoning stays locally competent. Reliable execution therefore requires an operating map, meaning a machine-usable representation of how a bounded unit of work is expected to progress, together with the context, checkpoints, and recovery paths that keep execution inside it. The practical management target is escaped material error, an error not prevented, detected, corrected, or escalated before material consequence. A Digital Operations System holds that map, localizes it to the organization, and improves it as the work is performed, while the underlying model remains unchanged.

Keywords: healthcare AI; operational reliability; escaped material error; Digital Operations; Digital Workers; Digital Labor; Digital Operations System; workflow state; operating map; localization; system training

Key points

- **Reliability belongs to the system.** Model capability is necessary but insufficient for dependable healthcare work, because the same model produces materially different results inside different operating environments.
- **Long-horizon work fails through state.** Once the workflow state is wrong, later steps can be performed competently and still be wrong, which is why the system needs an independent reference for correct progression.
- **Escaped material error is the management target.** A contained error demonstrates that controls functioned. Only errors that cross into material consequence should count against the design.
- **System training is a Digital Operations function.** The operating map, context, controls, allocation, recovery paths, and authority improve as work is performed, independently of model weights, and the resulting asset stays with the organization.

The error that nothing caught

The study of operational reliability in healthcare AI can be informed by the study of systemic failures in other industries. On 3 September 1989, a Boeing 737 operated by Varig left Marabá, in northern Brazil, on a short evening leg to Belém. Before departure, the captain consulted the company flight plan for the magnetic heading to the destination. It read 0270. The airline had recently moved to a four-digit notation with an implied decimal before the final digit, so the printed value meant 027.0 degrees, roughly north-northeast. The captain, who had been on leave when the notation changed, read it as 270 degrees (due west) and set his horizontal situation indicator accordingly. The first officer set the same value instead of deriving it independently.¹

The aircraft then flew west for nearly an hour, away from Belém and into the Amazon basin. Several cues should have disconfirmed the heading, but none did. The crew was flying toward the setting sun on a route that runs north, the

destination airport had no radar coverage, and that leg carried no charted airway to serve as an independent cross-check. When the flight management system indicated they had passed the destination, the crew turned, searched visually, and followed a misidentified river. The aircraft ran out of fuel and was put down in the forest. Most of the fifty-four people aboard survived.¹

The least instructive part of the account is that a capable professional made a mistake, since reliable systems are designed on the premise that competent operators will sometimes be wrong. The durable finding of safety research across industries, and of patient-safety and human-factors research in particular, is that outcomes are determined by the system surrounding the person rather than by the error itself.²⁻⁵ What mattered on Flight 254 was not the first error. It was that the error survived every subsequent opportunity for detection and correction.

Flight 254 is used here strictly as a systems-reliability illustration and not as an analogy for AI itself. It illustrates a system operating in a persistently incorrect state, where every subsequent action is internally consistent and externally wrong. Healthcare organizations are now delegating comparable long-horizon work to Digital Workers, meaning AI-enabled operational roles configured to perform bounded organizational work under defined authority, supervision, and evidence requirements. The reliability question for a Digital Worker is the one Flight 254 poses. What in the surrounding system would catch an error in long-horizon healthcare workflows before it escapes?

The management target is escaped material error

Healthcare AI reliability is often discussed as though the central question were whether the model makes mistakes. That framing is too blunt to manage healthcare workflows. This article explains why we should focus on eliminating escaped material errors, which are errors not prevented, detected, corrected, or escalated before they create a material operational, clinical, financial, legal, equity, reputational, or safety consequence. The qualifier *material* matters because

healthcare work includes minor imperfections, stylistic differences, and inconsequential inconsistencies.

The distinction changes how leaders should read failures. A Digital Worker that draws an incorrect intermediate inference which the system then catches is not evidence of unreliability. It may be the clearest available indication that the design is working. At the same time, fluent, accurate-looking output produced without any mechanism for detecting downstream state deviation can carry substantial risk.

Escaped material error is a management measure, not a substitute for accuracy, calibration, sensitivity, specificity, or other patient-safety standards. It focuses on consequential results rather than model predictions. Operational reliability aims not for zero internal errors but for preventing significant errors from escaping. This is achieved within a reliability envelope, which includes conditions under which the organization can demonstrate that a Digital Worker performs its tasks safely within defined limits. The reliability envelope specifies role scope, allowed sources, needed context, tool permissions, action boundaries, verification procedures, escalation points, supervisor roles, evidence requirements, and conditions for revocation. When Digital Worker actions fall outside this envelope, proper responses include verification, abstention, or escalation, instead of improvisation.

Improving the reliability of Digital Labor matters because administrative and documentation work drains clinician and staff capacity, contributes to burnout, and creates high avoidable cost.⁶⁻⁸ Prior authorization is repeatedly cited as a source of care delay and staff time, and independent analysis finds administrative AI already in wide use with uneven measured impact.^{9,10} Interoperability initiatives are making the data more machine-addressable, which helps Digital Workers reach the work without addressing the reliability problem described here.¹¹

Long-horizon work fails through incorrect states

The usual explanation of long-horizon failure is arithmetic. If a workflow contains many material steps and each carries an independent probability of success, end-to-end reliability decays roughly as the product of those probabilities. Fifty material steps at 99% reliability each complete cleanly about three times in five. Conversely, achieving 99% end-to-end reliability would require nearly 99.98% per-step accuracy. Analysis of frontier agent performance across task lengths supports that picture, fitting a constant failure rate per unit of human task time closely enough that each agent can be characterized by a half-life, with success declining exponentially as the number of steps per task increases.^{12,13}

The arithmetic still understates the challenge in healthcare because it treats steps as independent when healthcare work is stateful. A prior authorization is not a sequence of unrelated decisions. It identifies the patient and service, determines the payer and plan, identifies the governing policy, determines whether authorization is required, gathers documentation, assesses completeness, submits, monitors, responds to

requests for information, reconciles the outcome, and updates downstream systems. If the payer or governing policy version is determined incorrectly at the third step, everything after it can execute competently against a case already bound to the wrong rule. Sophisticated reasoning then faithfully executes the wrong workflow (**Fig. 1**).

A benchmark of computer-use agents on realistic administrative environments spanning an electronic health record, two payer portals, and a fax system reports strong subtask performance alongside low end-to-end completion, with the strongest configuration reaching 36.3% task success while the highest subtask success rate was 82.8%. Its authors attribute much of the gap to sustained state tracking over many steps.¹⁴ A larger simulation of managed-care operations found the best of thirty agent and harness configurations resolving 28.0% of tasks, no configuration clearing 20% when the same task had to pass three consecutive times, and performance falling to 3.8% when all cases in a domain were worked in one long session.¹⁵ After clinical judgment, the largest failure category was workflow completion, meaning a required terminal action was never taken.

A third evaluation, released in June 2026, makes the underlying condition explicit. It places frontier agents in fifty-four end-to-end healthcare tasks across seven distinct environments, spanning radiographs, computed tomography, whole-slide pathology, free text, and structured records. Each task supplies minimal instruction and requires the agent to explore raw data and construct a multi-step solution rather than answer a prompt. The strongest and most cost-effective agent reached roughly 42%, and tasks that combined large search spaces with compositional reasoning remained difficult for every agent tested.¹⁶ Earlier evaluation in virtual electronic health record environments and multi-turn rubric-based assessment pointed the same way, showing that averages conceal task-specific weakness and that performance must be measured in workflow-like conditions rather than in isolated knowledge tasks.^{17,18}

Read together, these evaluations describe a single condition rather than four separate results. In each case, a highly capable system enters a demanding environment with general competence, minimal instruction, and whatever it can infer from what it is given. That is close to the condition of the crew on Flight 254, who were skilled and fully equipped and working from a starting value that no one had independently checked. Capability is improving quickly, and these numbers will rise. What does not improve on its own is the lack of an independent reference for correct progression, the lack of local facts, and the absence of any supervisory mechanism that detects when the work has left the route. Healthcare is a regulated environment where errors can have significant consequences for patients, organizations, and society, and tolerance for AI-caused errors is expected to be low. This unavoidable high bar means we are unlikely to achieve reliable Digital Workers solely by improving intelligence capabilities. The system around the model will be essential in closing this gap.

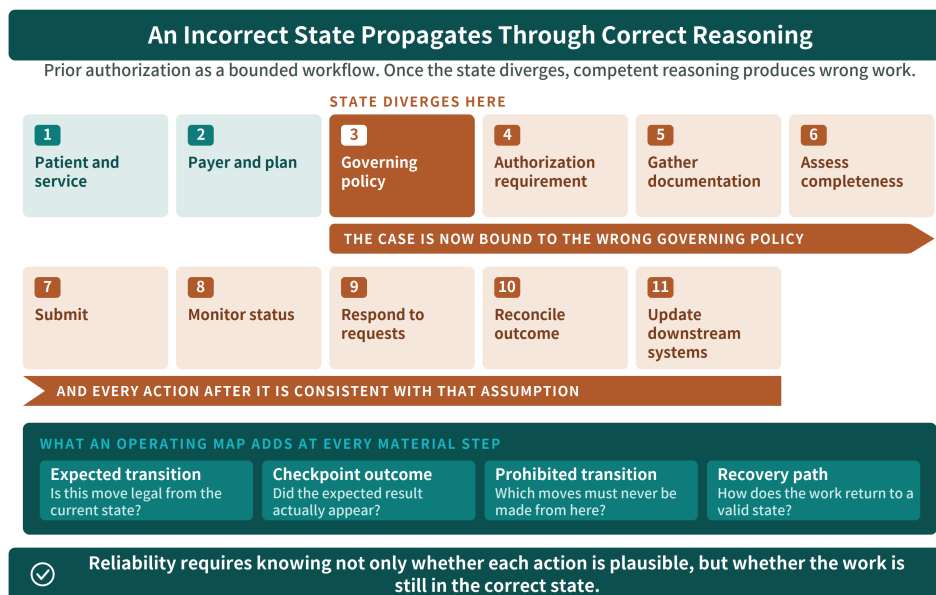


FIGURE 1 - An incorrect state propagates through correct reasoning. Prior authorization is a stateful workflow. Once the governing payer or policy is identified incorrectly, later steps can be executed competently against a case that is already bound to the wrong rule.

An operating map supplies the expected transitions, checkpoints, prohibited transitions, and recovery paths against which progression can be checked.

The operating map

If the system must know whether the work is in the correct state, something has to define what correct progression looks like. Call that representation an operating map, meaning a machine-usable statement of how a bounded unit of work is expected to progress. An operating map declares the work's canonical states and the material steps between them. It declares expected and prohibited transitions, the context each step requires, the outcome each step should produce, the exception states the work may legitimately enter, the conditions that require escalation, the paths by which the work returns to a valid state, and the evidence of completion.

The map is constructed from discrete units of work instead of workflow names, with each unit broken down into sequential steps. Each step involves exactly one type of work: retrieval, judgment, or action, never a combination of these, as they fail differently and require separate verification. Additionally, each step specifies the criteria for success, what occurs if that check fails, and the conditions that must be met for the work to stop and be handed to a person. Prohibitions are explicitly included as elements rather than implicitly left as the unspoken part of instructions, since an unmentioned act and an intentionally omitted act are different operational states that may appear identical in logs.

This matters because better reasoning cannot correct an incorrect operating state when the system holds no independent reference for what correct looks like from here. Intelligence can choose how to drive, but the operating map establishes where the road is, which direction is progress, and when the vehicle has left the route. Checkpoints, permissions, monitoring, and recovery then work more like lane markings, guardrails, and rerouting than like additional intelligence.

Comparing observed execution against a normative process model is an established discipline. Process mining separates discovery, conformance, and enhancement, and conformance checking pinpoints where real executions deviate from the model describing how the work should run.^{19,20} Healthcare has

applied it to clinical pathways and compliance monitoring for years. The difference here is one of timing rather than of concept. Process mining largely evaluates work that has already happened, while a Digital Operations System (DOps) has to use the same representation prospectively, while the work is still running, because the work system increasingly participates in deciding what happens next. The managed-care simulation encodes that structure, with case state machines carrying twenty-nine statuses and explicit legal transitions.¹⁵

Evidence from outside healthcare points the same way. The scaffolding that surrounds a model, often called the harness, accounts for a substantial share of measured agent performance. One command-line benchmark reports the same model at 52.1% under one harness and 57.8% under another while consuming 256.9 million against 3.9 million input tokens, an order-of-magnitude difference in operating efficiency at comparable capability.²¹ The managed-care simulation varied both model and harness across thirty configurations and found the best result in each domain split across different pairings rather than tracking a single strongest model.¹⁵ Models plainly matter, but a substantial part of what determines whether work is done reliably sits outside them.

Localization adapts the map to the organization

A canonical operating map can describe how bounded work should progress in general. It does not know how that map applies here, now, to this case. Closing that distance is localization, where general operating knowledge becomes the organization's own.

The distinction is clean in practice. It is canonical that authorization requires documentation supporting medical necessity. It is local which payer policy governs this patient, which documentation source is authoritative, which portal is used, what the submission deadline is, and who may approve an exception. The operational rule that follows is simple enough. Missing context is not permission to infer confidently, and missing, stale, conflicting, or low-authority context should trigger a control response rather than a best guess (**Table 1**).

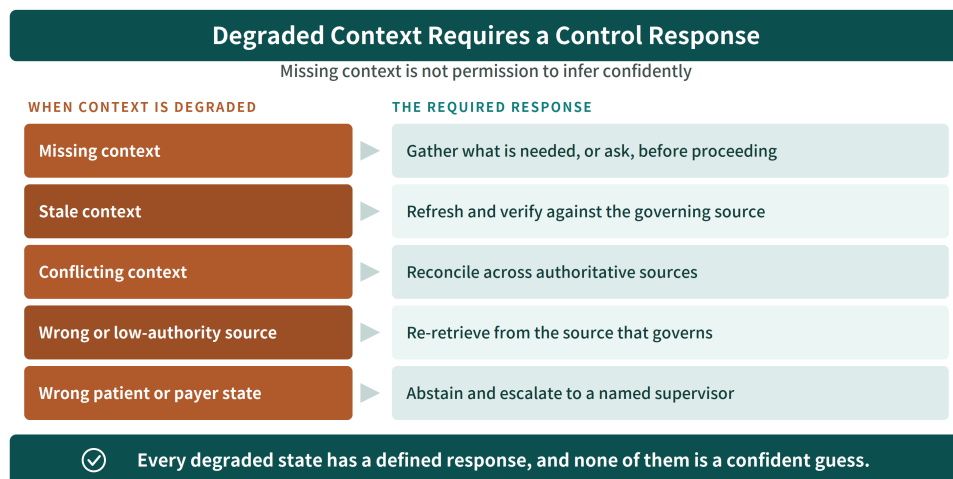


TABLE 1 - Degraded context and the required control response. Missing, stale, conflicting, or low-authority context should trigger verification, reconciliation, information gathering, abstention, or escalation rather than confident inference.

Many failures that look like reasoning failures are context failures. The system may hold the general clinical or administrative concept and still lack the local facts that determine the correct action, or may not know whether a source is authoritative, stale, superseded, or contradicted by another. Evidence from the managed-care simulation is direct. Removing the domain operations handbook from the agent's available knowledge cut utilization-management success from 32.0% to 17.3%. The same study found that a very large body of operating knowledge can also induce over-verification and refusal in other workflows, which is a useful caution against treating more context as automatically better.¹⁵ Enterprise practice has converged from a different direction, treating

organizational policies and standards as a machine-readable corpus that has to be engineered rather than assumed.²²

Keeping the two layers distinct is what makes the arrangement work. The canonical layer holds what good performance generally requires, including states, steps, checks, prohibitions, and exception categories. The local layer holds what only this organization can supply, including which payer policy governs, which system of record is authoritative, the submission deadline, and who may approve an exception. Local practice is captured as an instance of the canonical map rather than as an edit to it, so improvements to the canonical layer reach every organization that uses it, and one site's workaround never quietly becomes the general rule (**Fig. 2**).

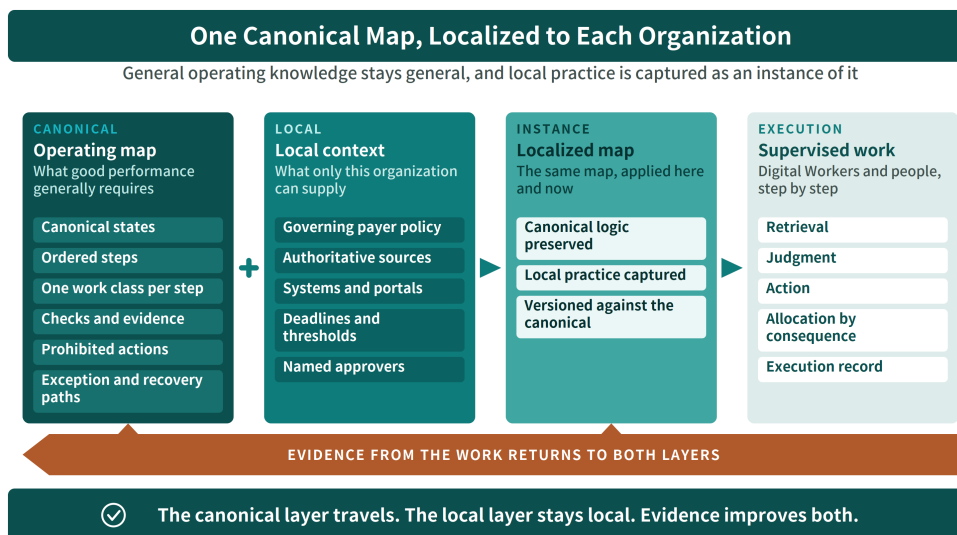


FIGURE 2 - One canonical operating map, localized to each organization. The canonical layer holds what good performance generally requires. Local context supplies what only the organization can know. The localized map governs supervised execution, and evidence from the work returns to both layers.

That separation is also what makes reliability infrastructure portable, and it opens a possibility worth naming carefully. Authoritative organizations such as health systems, specialty societies, professional organizations, payers, clinical networks, and standards bodies could develop, review, or qualify canonical operating knowledge for a defined domain. Qualified is not a single claim, and any such designation requires explicit criteria. At least five dimensions are separable. Semantic validity asks whether the map faithfully

represents the intended expert practice. Workflow validity asks whether the states, steps, dependencies, exceptions, and completion criteria are correct. Execution validity asks whether a Digital Worker can use the map to perform the bounded work reliably. Localization validity asks whether it can be instantiated elsewhere without violating its essential logic. Operational validity asks whether implementation improves escaped error, rework, cycle time, or burden.

Whoever writes it, operating knowledge has to be authored and maintained rather than generated. Benchmarks of machine-readable procedural knowledge find that automatically generated procedures typically provide no benefit, while curated procedures authored by people who know the work consistently outperform working without them.^{23,24} Quality is the variable that matters, and quality here is a function of how well the work was understood before it was written down.

Digital Operations runs the map

Digital Operations is the management discipline that begins when AI enters accountable work. Governance determines whether an AI use is acceptable and under what conditions, while Digital Operations manages the Digital Worker in operation, covering its role, organizational context, authority, supervision, evidence, incidents, improvement, and lifecycle. The Digital Operations System (DOPs) maintains those elements across Digital Workers.

The reliability argument adds an active dimension. A DOPs should maintain the current workflow state, the relevant organizational context, worker identity and authority, the expected progression, the checkpoints, the execution evidence, the deviations, the escalation routes, the recovery paths, and the current autonomy state. Described that way, it is less a governance archive than an operating system for the work. The claim needs a precise boundary, because such a system does not deterministically control every reasoning step. Probabilistic intelligence can remain probabilistic while material workflow transitions are governed by explicit state, evidence, permission, and authority conditions.

That boundary makes a five-part cycle operational rather than rhetorical (**Fig. 3**). **Prevent** blocks a material action that lacks required evidence or permission. **Detect** notices that an expected checkpoint outcome is absent or that a transition is invalid. **Contain** holds downstream state changes before consequence. **Recover** re-retrieves context, corrects state, reroutes the case, or escalates. **Learn** incorporates the correction into the context, the operating map, a control, or an authority decision. Telemetry answers what happened, while active reliability management asks whether the work is still progressing inside its expected envelope and what must happen now if it is not.

All of this depends on the record the work leaves behind. An execution record covering inputs, source versions, tool calls, intermediate decisions, uncertainty flags, policy versions, approvals, overrides, escalations, outcomes, and corrections lets an organization reconstruct what happened, locate where the error entered and where detection failed, separate model failure from context, workflow, tool, or control failure, judge whether autonomy remains justified, and improve the system. That last use is the one most often left unexercised. The same record later supports audit, contracting, and liability analysis,

where exposure turns substantially on organizational choices about selection, integration, validation, supervision, and reliance rather than on the model alone.²⁵

Who does what, surface by surface

Controls in healthcare AI often collapse into general language, including guardrails, review, governance, and monitoring. Those words are not wrong, but they are too coarse to manage work. The three work classes carried by each step in the operating map are the practical alternative. **Retrieval-dominant steps** ask whether the system obtained the right facts from the right source at the right time. **Judgment-dominant steps** ask whether those facts support a conclusion under the governing clinical, administrative, contractual, or legal standard. **Action-dominant steps** ask whether the system may safely change external state. Submitting to a payer, sending a patient message, scheduling an appointment, updating a queue, or writing to a record are not outputs. They are institutional acts.

The reliability implication is that task characteristics and consequences should determine human involvement, rather than applying it as a uniform workflow-level requirement. For every material step, an organization should be able to say what kind of step it is, what the consequence would be if it were wrong, what local evidence exists for Digital Worker performance on it, and what authority should therefore be delegated (**Table 2**).

That allocation replaces the indiscriminate application of a human-in-the-loop, which is not automatically protective. Automation-bias research has shown that people over-rely on automated recommendations when verification is difficult, the task is cognitively demanding, or the system appears authoritative.²⁶⁻²⁸ Work on algorithm-in-the-loop decision-making finds that human oversight frequently becomes procedural rather than substantive, and that deliberate friction is sometimes needed to restore genuine engagement.^{29,30} Classic human-factors work reached the same conclusion decades earlier, describing graded allocation of information acquisition, analysis, decision, and action rather than an all-or-nothing endpoint, and tying appropriate reliance to demonstrated system performance.^{31,32}

Early health-system deployments of AI-drafted patient messages illustrate the point in practice. Fluency alone did not determine value, which depended on editing burden, workflow fit, tone, and accountability for what was sent.³³ Contemporary framing treats intelligence as a property of the human-AI team rather than the model, shifting attention from model-centric validation to workflow-level design and continuous evaluation.^{34,35} People are therefore part of the operating system rather than an external safety layer applied uniformly to anything labeled AI.

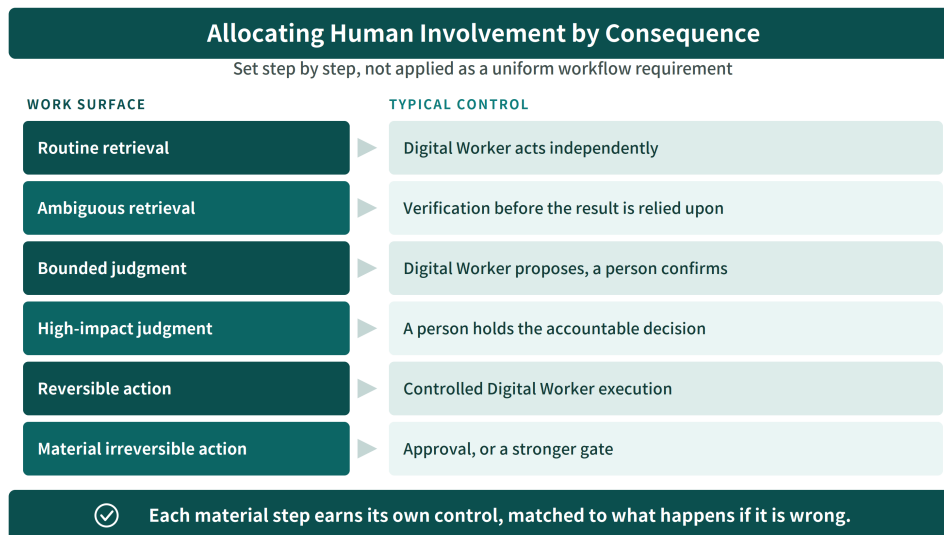


TABLE 2 - Allocating human involvement by consequence. Human review is set step by step, by task type, consequence, reversibility, uncertainty, and demonstrated performance, rather than applied as a uniform workflow-level requirement.

Autonomy is a system property

Autonomy is frequently described as a feature of the agent. However, we understand that it is safer to grant a Digital Worker authority for defined work under defined conditions. The same Digital Worker can hold different authority at different steps of the same workflow. It may retrieve claim status independently, propose an appeal for approval, act within a narrow envelope for low-risk updates, and be prohibited from high-consequence clinical communication.

Two consequences follow that are easy to get wrong. A model upgrade does not by itself justify expanded autonomy, because the evidence supporting the previous authority was collected on a different configuration. A model downgrade does not necessarily require full withdrawal, if the surrounding system compensates through tighter checkpoints, narrower action limits, or additional verification. The reliability envelope belongs to the whole arrangement of Digital Worker, operating map, context, controls, supervision, tools, and local environment, not to the model inside it. Authority should therefore expand, hold, narrow, or be revoked as evidence changes, and it has not been responsibly granted unless it can also be narrowed.

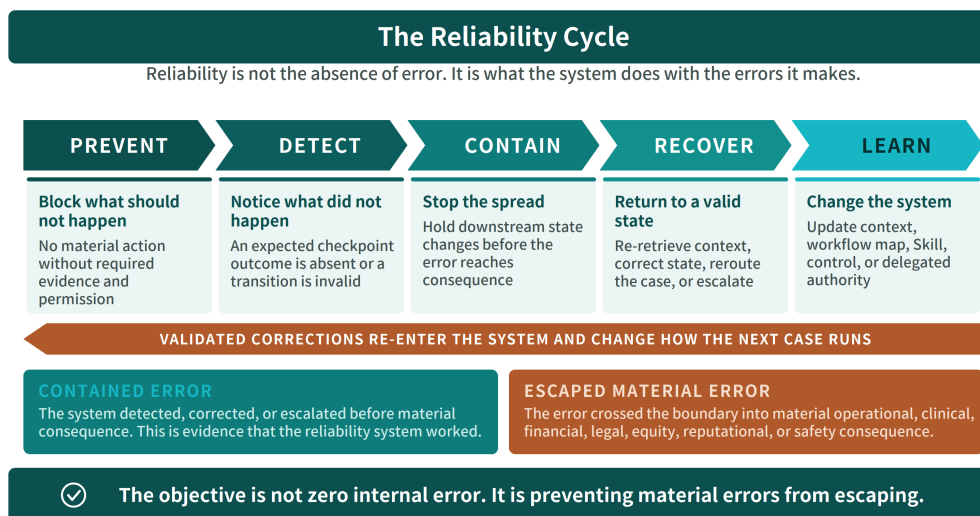


FIGURE 3 - The reliability cycle. Prevention, detection, containment, recovery, and learning are distinct operating functions, not a single monitoring activity. A contained error is evidence that the reliability system worked. Validated corrections re-enter the system and change how the next case is run.

System training improves the map

Model training optimizes parameters against data. System training improves the environment through which intelligence performs work, and it is an operating function rather than a project. A Digital Operations System (DOpS) that runs the map while the work executes is already positioned to improve it, because the corrections arrive as ordinary operating output rather than as findings from a separate review.

System training modifies the operating map, the context and source hierarchy, the checkpoints, the permissions, the escalation logic, the allocation of work between people and Digital Workers, the recovery mechanisms, and the authority granted at each step. The underlying model may remain entirely unchanged. That is the point, not a limitation, because it produces something the organization can retain, audit, version, carry across models, and improve through ordinary operations.

In practice, the loop runs as follows. The organization observes the work, detects a deviation, and classifies the failure as a context, workflow-state, map, permission, tool, handoff, or model-reasoning failure. The team makes the correction, updates the relevant component, validates the change, redeploys the work, and monitors for recurrence. Classification matters because only the last class implies anything about the model. Handled this way, error management stops being retrospective incident review and becomes continuous improvement of the operating system itself. A mature DOpS does not merely record failures. It converts corrections into improved operating intelligence, and a correction validated in one organization can strengthen the canonical layer that every other organization localizes.

The strategic reading follows without any claim about model plateaus. As general capability continues to improve and

becomes available across vendors, differentiation shifts toward the operating intelligence surrounding the model. Organizations generally do not own the frontier models they use. They can own the operating intelligence accumulated around them, and that operating intelligence survives the next model.

The executive deployment test

A sophisticated healthcare organization should not ask only whether an agent can perform a task in a demonstration. It should ask whether the organization can operate a Digital Worker as a dependable role. The questions below are a compact deployment test rather than a governance checklist (Fig. 4). They are the operating questions that determine whether capability has been converted into reliable Digital Labor.

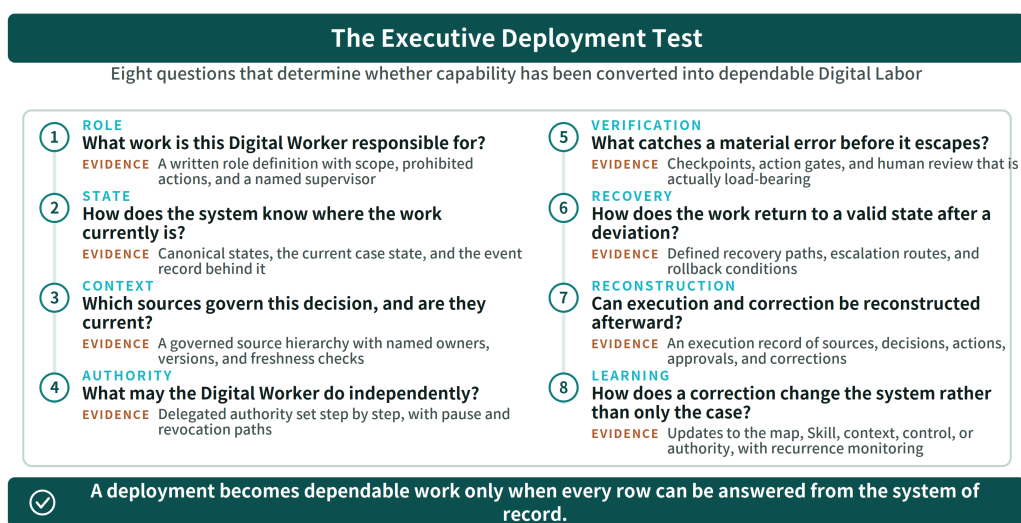


FIGURE 4 - The executive deployment test. Eight domains, each with the question an executive should be able to ask and the operating evidence a functioning Digital Operations System would produce.

Close: reliability is engineered

Building dependable healthcare AI means constructing work systems rather than acquiring autonomous agents and hoping model capability generalizes. Digital Workers need to operate against an explicit representation of how the work should progress, with current local context, controls matched to each work surface, deliberate allocation between people and Digital Workers, preserved evidence, recovery paths that return the work to a valid state, and authority granted only where local performance supports it.

This position does not depend on optimism or caution about model capability. It follows from how accountable work has to be operated. Capability precedes dependability, and dependability is earned when the organization engineers the conditions under which Digital Workers retrieve context, make bounded judgments, take actions, escalate uncertainty, recover from deviation, preserve evidence, and improve.

The strongest models a year from now will not be today's, and every organization in the market will be able to reach them. What will differ is whether an institution can state how its work is supposed to progress, detect when it is not, correct it, and keep the correction. That is a management asset, not a technical one. The organizations that pull ahead will not be the ones with the better model. They will be the ones with the better operating map, kept current by the work itself.

Glossary of key concepts

- **Operational reliability.** The demonstrated ability of an AI-enabled work system to complete a bounded healthcare function without escaped material error under defined local conditions.
- **Escaped material error.** An error not prevented, detected, corrected, or escalated before material operational, clinical, financial, legal, equity, reputational, or safety consequence.
- **Contained error.** An error the system detected, corrected, blocked, or escalated before material consequence. Evidence that the reliability design functioned, not that it failed.
- **Operating map.** A machine-usable representation of how a bounded unit of work is expected to progress, covering canonical states, ordered steps and their work class, checks, expected and prohibited transitions, required context, exception states, escalation conditions, recovery paths, and completion evidence.
- **Work class.** The single category carried by each step of an operating map. Retrieval, judgment, and action fail differently, are verified differently, and carry authority separately.
- **Localization.** The instantiation of a canonical operating map for one organization through its local context, systems, sources, thresholds, and named approvers, without altering the canonical layer.

- **Reliability envelope.** The set of conditions under which the organization has evidence that a Digital Worker can perform a bounded function safely enough for the authority granted.
- **Context failure.** A failure caused by missing, stale, conflicting, low-authority, or incorrectly prioritized local information rather than by lack of general capability.
- **System training.** The continuous improvement of the environment through which intelligence performs work, including the operating map, context, controls, allocation, recovery paths, and authority, carried out as a Digital Operations function while the underlying model may remain unchanged.

Live Article information

This PDF is a dated snapshot of a continuously maintained Live Article. Material updates undergo editorial review, and substantive public revisions are released when warranted.

The current maintained article and evidence base are available through the ConverGen Digital Operations Advisor: www.convergen.ai/advisor/

Peer-review status: This public edition has not undergone external peer review.

Competing interests: The author is the founder and CEO of ConverGen AI, which develops and commercializes a Digital Operations System (DOPS) for healthcare organizations.

AI-assisted tools: AI-assisted tools supported research synthesis, drafting, editing, and figure production. The author reviewed and takes responsibility for all content.

Author ORCID: 0009-0008-1814-1983

References

1. Aviation Safety Network. Accident description: Boeing 737-241 PP-VMK, Varig flight 254, 3 September 1989. Flight Safety Foundation. <https://aviation-safety.net/wikibase/326246>. Accessed 28 Aug 2026.
2. Reason, J. Human error: models and management. *BMJ* 320, 768–770 (2000). <https://doi.org/10.1136/bmj.320.7237.768>.
3. Kohn, L. T., Corrigan, J. M. & Donaldson, M. S. (eds). *To Err Is Human: Building a Safer Health System*. National Academies Press (2000). <https://doi.org/10.17226/9728>.
4. Carayon, P. et al. Work system design for patient safety: the SEIPS model. *Qual. Saf. Health Care* 15 Suppl. 1, i50–i58 (2006). <https://doi.org/10.1136/qshc.2005.015842>.
5. Salwei, M. E. & Carayon, P. A sociotechnical systems framework for the application of artificial intelligence in health care delivery. *J. Cogn. Eng. Decis. Mak.* 16, 194–206 (2022). <https://doi.org/10.1177/15553434221097357>.
6. Sinsky, C. et al. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. *Ann. Intern. Med.* 165, 753–760 (2016). <https://doi.org/10.7326/M16-0961>.
7. Shrank, W. H., Rogstad, T. L. & Parekh, N. Waste in the US health care system: estimated costs and potential for savings. *JAMA* 322, 1501–1509 (2019). <https://doi.org/10.1001/jama.2019.13978>.
8. Han, S. et al. Estimating the attributable cost of physician burnout in the United States. *Ann. Intern. Med.* 170, 784–790 (2019). <https://doi.org/10.7326/M18-1422>.
9. American Medical Association. 2025 AMA Prior Authorization Physician Survey (2025). <https://www.ama-assn.org/system/files/prior-authorization-survey.pdf>. Accessed 15 Jul 2026.
10. Peterson Health Technology Institute. Administrative AI: Current Use and Potential Impact (2026). <https://phti.org/administrative-ai-current-use-and-potential-impact/>. Accessed 15 Jul 2026.
11. Centers for Medicare & Medicaid Services. 2024 CMS Interoperability and Prior Authorization Final Rule (CMS-0057-F) (2024). <https://www.cms.gov/initiatives/burden-reduction/overview/interoperability/policies-regulations/cms-interoperability-prior-authorization-final-rule-cms-0057-f>. Accessed 30 Jun 2026.
12. Ord, T. Is there a half-life for the success rates of AI agents? Preprint at <https://arxiv.org/abs/2505.05115> (2025). <https://doi.org/10.48550/arXiv.2505.05115>.
13. Kwa, T., West, B. et al. Measuring AI ability to complete long tasks. Preprint at <https://arxiv.org/abs/2503.14499> (2025). <https://doi.org/10.48550/arXiv.2503.14499>.
14. Bedi, S. et al. HealthAdminBench: evaluating computer-use agents on healthcare administration tasks. Preprint at <https://arxiv.org/abs/2604.09937> (2026). <https://doi.org/10.48550/arXiv.2604.09937>.
15. Chen, H. et al. x-Bench: can AI agents automate end-to-end, long-horizon, policy-rich healthcare workflows? Preprint at <https://arxiv.org/abs/2605.16679> (2026). <https://doi.org/10.48550/arXiv.2605.16679>.
16. Liu, Q. et al. HealthAgentBench: a unified benchmark suite of realistic agentic healthcare environments for challenging frontier AI agents. Preprint at <https://arxiv.org/abs/2606.31179> (2026). <https://doi.org/10.48550/arXiv.2606.31179>.
17. Jiang, Y. et al. MedAgentBench: a virtual EHR environment to benchmark medical LLM agents. *NEJM AI* 2, AIdbp2500144 (2025). <https://doi.org/10.1056/AIdbp2500144>.
18. Arora, R. K. et al. HealthBench: evaluating large language models towards improved human health. Preprint at <https://arxiv.org/abs/2505.08775> (2025). <https://doi.org/10.48550/arXiv.2505.08775>.
19. van der Aalst, W. M. P. *Process Mining: Data Science in Action*. 2nd edn. Springer (2016). <https://doi.org/10.1007/978-3-662-49851-4>.
20. Process mining applications in healthcare: a systematic literature review. *PeerJ Comput. Sci.* 11, e2613 (2025). <https://doi.org/10.7717/peerj-cs.2613>.
21. Merrill, M. A. et al. Terminal-Bench: benchmarking agents on hard, realistic tasks in command line interfaces. Preprint at <https://arxiv.org/abs/2601.11868> (2026). <https://doi.org/10.48550/arXiv.2601.11868>.
22. Context engineering: from prompts to corporate multi-agent architecture. Preprint at <https://arxiv.org/abs/2603.09619> (2026). <https://doi.org/10.48550/arXiv.2603.09619>.
23. Li, X. et al. SkillsBench: benchmarking how well agent skills work across diverse tasks. Preprint at <https://arxiv.org/abs/2602.12670> (2026). <https://doi.org/10.48550/arXiv.2602.12670>.
24. MIND-Skill: quality-guaranteed skill generation via multi-agent induction and deduction. Preprint at <https://arxiv.org/abs/2605.08670> (2026). <https://doi.org/10.48550/arXiv.2605.08670>.
25. Mello, M. M. & Guha, N. Understanding liability risk from using health care artificial intelligence tools. *N. Engl. J. Med.* 390, 271–278 (2024). <https://doi.org/10.1056/NEJMhle2308901>.
26. Goddard, K., Roudsari, A. & Wyatt, J. C. Automation bias: a systematic review of frequency, effect mediators, and mitigators. *J. Am. Med. Inform. Assoc.* 19, 121–127 (2012). <https://doi.org/10.1136/amiajnl-2011-000089>.
27. Lyell, D. & Coiera, E. Automation bias and verification complexity: a systematic review. *J. Am. Med. Inform. Assoc.* 24, 423–431 (2017). <https://doi.org/10.1093/jamia/ocw105>.
28. Khera, R., Simon, M. A. & Ross, J. S. Automation bias and assistive AI: risk of harm from AI-driven clinical decision support. *JAMA* 330, 2255–2257 (2023). <https://doi.org/10.1001/jama.2023.22557>.
29. Green, B. & Chen, Y. The principles and limits of algorithm-in-the-loop decision making. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, 50 (2019). <https://doi.org/10.1145/3359152>.
30. Buçinca, Z., Malaya, M. B. & Gajos, K. Z. To trust or to think: cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, 188 (2021). <https://doi.org/10.1145/3449287>.
31. Parasuraman, R., Sheridan, T. B. & Wickens, C. D. A model for types and levels of human interaction with automation. *IEEE Trans. Syst. Man Cybern.* A 30, 286–297 (2000). <https://doi.org/10.1109/3468.844354>.
32. Lee, J. D. & See, K. A. Trust in automation: designing for appropriate reliance. *Hum. Factors* 46, 50–80 (2004). <https://doi.org/10.1518/hfes.46.1.50.30392>.
33. Garcia, P. et al. Artificial intelligence-generated draft replies to patient inbox messages. *JAMA Netw. Open* 7, e243201 (2024). <https://doi.org/10.1001/jamanetworkopen.2024.3201>.
34. Prioritizing human-AI collaboration in healthcare: the TRIAD framework for trustworthy governance, real-world value, and integrated adaptive deployment. *Mil. Med. Res.* (2026). <https://doi.org/10.1186/s40779-026-00684-w>.
35. Operationalizing trustworthy artificial intelligence in clinical and operational workflows. *Front. Digit. Health* 8, 1779041 (2026). <https://doi.org/10.3389/fdgh.2026.1779041>.