Accuracy Is Not Reliability: Engineering Agent Evaluation Beyond Static Benchmarks

Exzil Calanza

Independent Researcher, Iloilo City, Philippines

Independent research preprint. IEEE-style formatting only; this work has not been accepted by, published by, certified by, or peer reviewed by IEEE.

ABSTRACT This technical synthesis examines why improvements on static accuracy benchmarks can fail to translate into reliable agent behavior under operational perturbations. Using the source article's cited research on semantic rephrasing, tool-use failure, robustness, and verification and validation, it distinguishes capability measurement from reproducible system reliability. The paper relates those findings to software verification and testing standards as engineering references, not as evidence of formal standards compliance. It presents no novel trial dataset; its contribution is a source-grounded reliability framing and a set of repeatable evaluation controls for agentic systems.

INDEX TERMS ¢â‚¬â€□ agent reliability; benchmark accuracy; robustness; software verification and validation; engineering standards; chaos testing

I. INTRODUCTION

STUDY TYPE: Technical synthesis. **RESEARCH QUESTION:** How do operational stress conditions, semantic perturbations, and tool failures differentiate agent reliability from static benchmark accuracy?

This preprint formalizes the existing public evidence record as a technical synthesis. No new experiment, dataset, measurement, or validation is claimed beyond the source article and its cited evidence.

The score went up and the agent still broke. Two 2026 papers show why benchmark accuracy and operational reliability are not the same property.

II. KEY TAKEAWAYS

- Capability and reliability are different properties. Fifteen models measured across twelve reliability metrics showed recent capability gains yielding only small reliability improvements.
- A rephrase cost roughly one success in eleven. Under semantically equivalent perturbations, success fell from 96.9% to 88.1% — and the task never got harder.
- "Just upgrade the model" does not settle it. The perturbation result does not separate model robustness from the scaffolding around it, and a single accuracy number tells you neither.
- Most quoted reliability statistics do not trace. Several widely repeated figures were dropped from this piece because the original either could not be retrieved or measured something different.

Ask why a production AI agent failed and you will usually be shown a benchmark score. The score went up; the agent still broke. That is not a paradox, and it is not a mystery. It is a measurement error that the field has been making at scale: **benchmark accuracy and operational reliability are different properties, and improving the first does not reliably improve the second.** Two papers published this year make that concrete enough to argue with.

III. CAPABILITY IMPROVED. RELIABILITY BARELY MOVED.

In *Towards a Science of AI Agent Reliability* (arXiv:2602.16666), Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala and Arvind Narayanan start from the observation that "rising accuracy scores on standard benchmarks suggest rapid progress, while many agents still continue to fail in practice."

Their response is to stop treating reliability as one number. They propose twelve metrics across four dimensions — consistency, robustness, predictability and safety

— and evaluate fifteen models against them. The finding that matters for anyone choosing a model on a leaderboard: "recent capability gains have only yielded small improvements in reliability."

Read that as a purchasing warning. If your selection criterion is a single success-rate number, you are optimising a variable that has been improving while the one you actually care about has not moved much.

IV. THE FAILURE MODE IS FRAGILITY, NOT INCOMPETENCE

ReliabilityBench: Evaluating LLM Agent Reliability Under Production-Like Stress Conditions (arXiv:2601.06112, January 2026) attacks the same gap from the other end: what happens when you stop giving the agent clean inputs?

The benchmark tests three things production has and benchmarks usually do not — consistency when a task repeats, robustness when a task changes slightly, and resilience when a tool fails. It uses chaos-engineering techniques, injecting faults like timeouts and rate limiting, across scheduling, travel, customer support and e-commerce domains.

The headline number is the one to sit with. Under semantically equivalent perturbations — the task means the same thing, phrased differently — success falls from **96.9% at ε=0 to 88.1% at ε=0.2**.

Nothing about the task got harder. The agent was asked to do the same thing in slightly different words, and roughly one in eleven successes disappeared — 8.8 points off a 96.9% base is about 9% of the successes it previously had. A 96.9% system and an 88.1% system are very different products, and the difference between them is not capability. It is brittleness that the clean benchmark never exposed.

V. WHY THIS DISTINCTION IS EXPENSIVE

An 8.8-point drop sounds survivable until it compounds. Production agents rarely do one thing; they chain tool calls, and each step is another opportunity for the phrasing, the state or the tool to differ from the benchmark's idealised conditions.

A per-step reliability figure and an end-to-end workflow reliability figure are not the same number, and the gap widens with every step in the chain.

This is also why "just upgrade the model" is a weaker move than it sounds. It assumes end-to-end behaviour is a function of model quality alone. The perturbation result complicates that: the same model, on the same task, produced different outcomes for semantically equivalent phrasings.

What that experiment does *not* settle is where the fix belongs. Paraphrase brittleness could be a property of the model's own robustness, of the scaffolding around it — input handling, state assumptions, recovery paths — or of both, and the measurement as reported does not separate them. That is worth stating plainly rather than resolving in whichever direction suits the argument. What it does establish is that a single clean accuracy number will not tell you which one you are dealing with.

VI. THE HONEST COUNTER-ARGUMENT

The strongest objection is that none of this has stopped agents from being useful. Organisations do report productivity gains from deployed agents, and a reliability ceiling measured in a benchmark is not the same as a business failing to get value.

Both things can be true: agents can produce real returns *and* have a reliability profile their accuracy scores do not describe. The mistake is not deploying them; it is deploying them against the wrong number.

VII. A NOTE ON THE NUMBERS IN THIS ARTICLE

There is a second-order problem worth naming. This field's most-quoted statistics — the share of pilots that never reach production, the percentage of agent projects predicted to be cancelled — circulate widely, are usually attributed to a major research house, and frequently cannot be traced back to a primary publication that says what the citation claims. Several figures considered for this article were dropped for exactly that reason: the number was repeated everywhere and the original either could not be retrieved or measured something different from what the quotation asserted.

So this piece cites two papers, both retrieved and read directly, and declines to launder the rest. If a reliability statistic matters enough to base a decision on, it matters enough to open the source.

VIII. WHAT TO MEASURE INSTEAD

Three questions that survive contact with production, drawn from what these papers actually test:

1. Does it repeat? Run the same task many times. A single success is an anecdote; the distribution is the property.
2. Does it survive a rephrase? Semantically equivalent inputs should produce equivalent outcomes. When they do not, you have measured brittleness directly.
3. What happens when a tool fails? Timeouts and rate limits are normal operating conditions, not edge cases. An agent that has never been tested against a failing dependency has not been tested.

None of these require a new model. They require deciding that reliability is a thing you measure on purpose, rather than a thing you infer from an accuracy score that was never designed to tell you about it.

IX. RELIABILITY NEEDS A VERIFICATION-AND-VALIDATION PROTOCOL, NOT A SLOGAN

A stronger reliability claim has to be tied to a repeatable procedure. IEEE 1012-2024 defines verification and validation (V&V) as processes for determining whether development products conform to their requirements and whether the resulting system satisfies intended use and user needs [4]. ISO/IEC/IEEE 29119-2:2021 likewise specifies test processes for governing, managing and implementing software testing across lifecycle models [5]. Those standards do not provide an AI-agent benchmark score, but they do establish a useful discipline: define requirements, define evidence, execute tests, preserve results, and separate verification from assertion.

For an agent workflow, that suggests a minimum experiment: freeze the task and tool versions; run repeated trials; add semantically equivalent paraphrases; inject dependency failures such as timeouts and rate limits; record every tool call and externally observable result; and report both task-level success and end-to-end workflow success with sample size. A result without the run conditions, failure taxonomy and evidence trail should be treated as an observation, not a general reliability claim.

X. STANDARDS ALIGNMENT IS NOT STANDARDS COMPLIANCE

This article does **not** claim that the engineering controls discussed here establish IEEE, ISO or NIST certification or full standards compliance. The narrower claim is that several engineering controls can be evaluated against established software-assurance concepts. IEEE 730-2026 establishes requirements for software quality assurance processes and is harmonized with ISO/IEC/IEEE 12207:2026 lifecycle processes [3][6]. IEEE 1012-2024 covers V&V across systems, software and hardware [4]. ISO/IEC/IEEE 29119-2:2021 defines software test processes [5].

For AI-specific risk, NIST AI RMF 1.0 is explicitly voluntary and organizes risk work around GOVERN, MAP, MEASURE and MANAGE [7]. NIST's Generative AI Profile extends that framework for generative-AI risks [8]. NIST SP 800-218 defines the Secure Software Development Framework, while SP 800-218A adds AI-model-specific secure-development practices [9][10]. OWASP's Agentic AI guidance adds a threat-model view for autonomous systems [11]. These are references for design and audit; they are not badges that can be claimed from a checklist.

XI. WHAT THIS CHANGES IN AGENT ENGINEERING

The engineering target is now stricter than "the tool returned success." A completion claim should carry enough evidence for an independent checker to distinguish at least four states: *requested*, *attempted*, *observed*, and *verified*. Mutating actions need post-condition evidence from the system that owns the state, not merely a click event or a local 200 response. Identity-sensitive browser work should fail closed when the requested account cannot be live-verified. Test reports should include the exact build, environment, sample count and failure class rather than collapsing everything into one pass percentage.

These controls are hypotheses to test, not proof that an agent system is reliable. Their value is falsifiability: if the post-condition cannot be observed, the run remains uncertain; if the account identity does not match, the action does not proceed; if a dependency failure is injected and recovery fails, the failure is recorded instead of being converted into a green receipt.

XIII. LIMITATIONS AND CLAIM BOUNDARY

The paper is a literature synthesis and analysis of secondary benchmark results. It presents no novel trial dataset and does not establish formal IEEE, ISO, or IEC compliance or certification.

Claim boundary - Explicitly disclaim formal IEEE/ISO/IEC compliance or certification and use standards only as engineering references.

XIV. REFERENCES AND SOURCE RECORD

1. [1] S. Rabanser et al., Towards a Science of AI Agent Reliability , arXiv:2602.16666, 2026. arXiv .
2. [2] ReliabilityBench: Evaluating LLM Agent Reliability Under Production-Like Stress Conditions , arXiv:2601.06112, 2026. arXiv .
3. [3] IEEE Std 730-2026, IEEE Standard for Software Quality Assurance Processes , 2026. IEEE .
4. [4] IEEE Std 1012-2024, IEEE Standard for System, Software, and Hardware Verification and Validation , 2024. IEEE .
5. [5] ISO/IEC/IEEE 29119-2:2021, Software and systems engineering ? Software testing ? Part 2: Test processes , 2021. ISO .
6. [6] ISO/IEC/IEEE 12207:2026, Systems and software engineering ? Software life cycle processes , 2017. ISO .

7. [7] E. Tabassi, Artificial Intelligence Risk Management Framework (AI RMF 1.0) , NIST AI 100-1, 2023, doi:10.6028/NIST.AI.100-1. NIST .
8. [8] C. Autio et al., Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile , NIST AI 600-1, 2024, doi:10.6028/NIST.AI.600-1. NIST .
9. [9] M. Souppaya, K. Scarfone and D. Dodson, Secure Software Development Framework (SSDF) Version 1.1 , NIST SP 800-218, 2022, doi:10.6028/NIST.SP.800-218. NIST .
10. [10] H. Booth et al., Secure Software Development Practices for Generative AI and Dual-Use Foundation Models: An SSDF Community Profile , NIST SP 800-218A, 2024. NIST .
11. [11] OWASP GenAI Security Project, Agentic AI ? Threats and Mitigations , 2025. OWASP .

Primary standards/framework pages and the two agent-reliability papers were rechecked on 10 August 2026. The standards are used here as engineering references; no certification or formal conformance claim is made.

AUTHOR BIOGRAPHY

EXZIL CALANZA is an independent researcher and software builder based in Iloilo City, Philippines. His work focuses on browser and desktop automation, reproducible evaluation, agent reliability, workflow verification, and AI-assisted systems engineering.