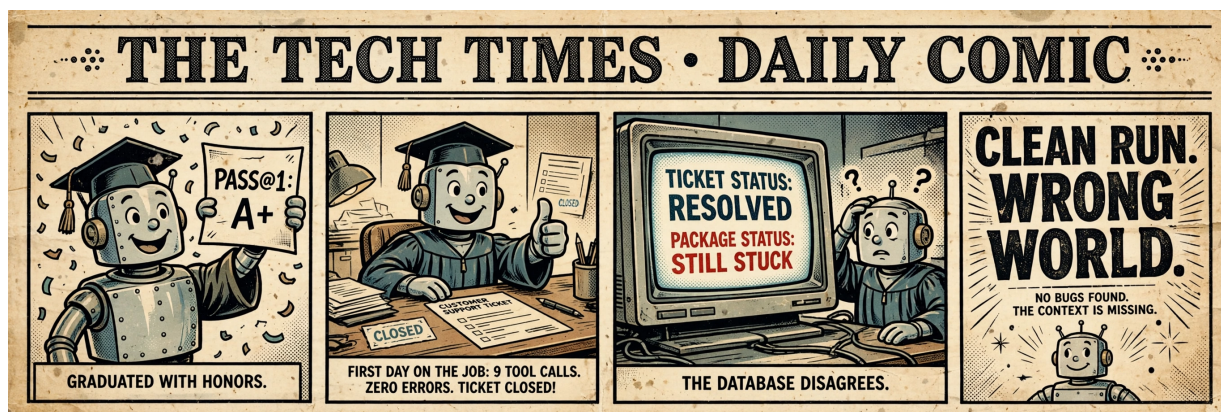


Stop Grading Agents Like Chatbots.

BY INSTINCTIVE NETWORK · OCTOBER 4, 2026



Clean termination. Wrong world. Illustration: Instinctive Network.

A customer writes in about a \$745 kitchen appliance stuck at a Nashville distribution center, fifteen days past its delivery date. The AI agent does careful work: nine tool calls. It pulls the order, checks tracking, reads the refund policy twice, confirms her account genuinely doesn't qualify for late-delivery compensation. Then it closes her ticket as **resolved** — and asks, brightly, "Is there anything else I may assist you with?"

Two things are wrong, and neither is in the tool calls. The package is still stuck, so the ticket should have been left "on hold," not marked resolved. And the customer never got an answer to what she actually asked.

This is the opening example from ThinkingBox, a Microsoft benchmark for stateful business agents featured in a joint Microsoft–Hugging Face post on October 3: 507 stateful business workflows across retail, auto insurance, travel, neobank, and consulting, each run 20 times and graded primarily on terminal database state and side effects rather than the agent's own account of what it did. Not the tool calls. Not the agent's summary. In the authors' words: "A trajectory is a claim. Database state is the evidence. Repetition is the trust test."

Here's what the evidence says. In a common-set ablation of 121,680 trials across 12 models, 79,853 attempts failed the executable checks — and 67.24% of those failures still "terminated cleanly, invoked a state-changing tool, and reported no final tool error." **Clean termination. Wrong world:** the state checks found wrong field values in 77.61% of them and unintended extra side effects in 43.30%.

Two out of three failures produced no explicit execution failure. The run ended normally. The world was still wrong.

That is a different problem from having a bad LLM judge. A judge can inspect whether an action looks sensible, whether a tool call is well formed, or whether the agent's explanation sounds compliant. None of those facts establishes that the intended state transition actually occurred without collateral effects.

A tool call is not an outcome.

Repetition makes it worse. Kimi-K3, the strongest open-weights model, solves 476 of 507 tasks at least once (93.89%) — but only 68 on all 20 attempts. Claude Opus 5.5 leads the single-attempt ranking at 67.16%, yet passes exactly the same 241 tasks 20-for-20 as the older Opus 5. "In this benchmark, a newer model does not fix the reliability problem."

And a successful trajectory is not a reliable system.

The eval industry is still grading agents with a chatbot's ruler. In the chatbot world, prompt in, response out, the response was usually the thing you cared about. Agents are different. They act on systems, change records, trigger side effects, and leave behind state. Between a plausible trajectory and a job actually done sits an entire layer that response-based evaluation can miss: **what changed in the world**. Many agent evals still inherit a chatbot-era assumption: that inspecting the model's output or trajectory is enough. The bill comes due in the latter.

So the number most leaderboards publish — pass@1 — **answers a useful question. Just not the reliability question.** It tells you how often the agent succeeds on an attempt. It does not tell you whether the same workflow will succeed reliably when repeated. The authors' advice is blunt: check the terminal state before you commit, not the model's summary of it.

(ThinkingBox exposes the measurement gap. We think that gap has an architectural consequence: agent systems need a verification layer that evaluates the state they leave behind, not just the actions they attempt. That is the problem our Measure Impact layer is built around: don't stop at whether the agent chose a plausible action — verify what actually changed.)

Don't evaluate what the agent says it did. Don't even stop at what action it took. **Evaluate what changed in the world.**

This is the principle behind Measure Impact: verify the state an agent leaves behind, including intended changes, missing changes, and unintended side effects.

Your agent's eval is green. What, exactly, did it measure?

