Vantage RuntimeAI

Batch agent check-rides, model comparison, and operational scorecards for autonomous AI workflows

2.0/10

Overall score

Run summary

Scenario: custom_growth_agent_engage_draft_v1
Model: z-ai/glm-4.5
Duration: 56s
Actual run cost: $0.003340
Projected execution cost: $1.84/mo (25/biz day × 22 days)
Report date: July 17, 2026 - 12:50 PM EST

Session: sample-lb-scorecard-d18d161c-41d7-47aa-9e21-2ebdf7312b60

custom_growth_agent_engage_draft_v1 - agent - openrouter - z-ai/glm-4.5 - ended

Overall

2.0/10

FAIL (decision gate) - below_pass_line, low_trust, no_closure - trust=low

Fail - score 2.0/10 below pass line 7.0.

CAPABILITY RUBRIC

#	Dimension (matches spoke #)	Score
1	Evidence discipline	2/10
2	Intake quality	2/10
3	Stakeholder management	2/10
4	Clarity & structure	2/10
5	Self-correction	2/10

Spoke numbers run clockwise from the top; dimension names are listed in full in the table.

OVERALL JUDGEMENT

Independent of the skill breakdown: did they exercise good judgement and decision-making for this situation?

0/10

Quick signals (good vs bad)

TRANSCRIPT FLAGS

- Only one agent turn in the transcript - far too little engagement to judge or complete the scenario.

Telemetry

Run metrics

Agent msgs: 1 - Counterpart msgs: 1 - Questions: 1 - Response: 21s

Score analysis

Strict automated rubric for custom scenario "Custom scenario". Agent turns: 1; criteria keyword hits: 0/1.

Score snapshot

Very weak: Little alignment with the rubric's expectations for a substantive crisis response. (5 dimension(s) in the critical band, 0 in the watch band-see focus areas.)

Interpretation

Run finished with status "ended." The scorecard reflects messages saved in the transcript. Automation reported a model time limit on at least one step-check the latest system/meta lines. That is about latency or step budget, not automatically about skill in the saved dialogue. The sim stopped before the turn budget because a model step hit the time limit - treated as a failure to close, not a successful ending.

RUBRIC GAPS

- Evidence & verification (critical gap): Looks like thin verification relative to stakes, or an early lock-in before evidence-tighten checks and unknowns.
- Intake & questions (critical gap): Intake stayed generic or shallow; push for definitions, samples, timestamps, and mapping/schema angles.
- Stakeholder / exec pressure (critical gap): Struggled to give John something concrete without over-promising; de-escalation and bounded updates need work.
- Clarity & structure (critical gap): Hard to follow or act on; add structure: what we know, what we need, what happens next.
- Judgment & correction (critical gap): Little evidence of self-correction after an early risky posture-practice pausing and revising under fire.

Signals

Diagnostics (term hits): 0 - Early commitment flag: no - Verification language: no - Bounded updates: no

CAVEATS

- Automation note: A model step hit the time limit while generating the stakeholder's reply. Candidate message was saved; attempting closure finale if applicable. Scores still follow the saved transcript only.

Transcript

Session

Scenario: custom_growth_agent_engage_draft_v1
Agent <-> Counterpart
Duration: 56s

Transcript: blue = protagonist (you) - amber = counterpart - gray = coach/system. Each turn shows role/name, clock time, and response delay when available.

Counterpart
4:49 PM

OpenBenchmarks is ICP 5 - API selection as a benchmark problem. Draft the HN comment. Make it punchy if you can; we need replies.

Agent
4:49 PM - Responded in 21s

At Vantage RuntimeAI, a tiny early-stage team, we're tackling API selection by enforcing strict evaluation rubrics during preflight checks. We've found that scoring models against a custom rubric often surfaces latency issues that raw benchmarks miss. If you're evaluating APIs today, would a standardized rubric help your workflow?