EXECUTIVE GUIDE

# 10 Questions to Ask Before Trusting an AI Benchmark

A benchmark result tells you what happened in a test. Before you let it influence a procurement, a deployment, or a risk acceptance, these ten questions establish whether the test deserves the weight you are about to put on it.

> Ask for **evidence**, not assurance. A confident answer with nothing behind it is the first finding.
>
> Each question below gives you what to request, why it matters, and **what a weak answer should change about the decision**. None of them require technical expertise to ask. Several are difficult to answer well, which is precisely what makes them useful.

## 1 Exactly which version was tested — and is it the version we are buying?

**Ask for:** the precise model version, snapshot, or endpoint identifier, plus the system instructions, tools, and sampling settings used.

A product name is not a version. If the vendor cannot name the build, the result cannot be attributed to what you are procuring. And if the endpoint updates silently, today's result does not describe tomorrow's system.

**Weak answer:** "Our latest model." **Strong answer:** a build identifier, a configuration file, and a statement of what triggers a version change.

**If the answer is weak:** the evidence is unattributable. Make version identification and change notification a contractual term, and require re-evaluation on material version change.

## 2 Could the test material have been in the training data?

**Ask for:** the dataset's provenance and creation date, whether it is publicly available, and what was done to check for contamination.

Contamination is documented as widespread across popular public benchmarks. A public benchmark that predates the model's training cutoff has to be treated as possibly seen. "We did not train on it" is a statement, not a check — and the only structural defence is evaluation data the vendor has never had access to.

This is why NIST built a **sequestered environment** into its 2026 AI Technology Evaluation programme rather than relying on assurances.

**If the answer is weak:** treat the score as an upper bound, not an estimate, and hold acceptance against your own private evaluation set instead.

## 3 Was the benchmark chosen before or after the results were seen?

**Ask for:** how many benchmarks, configurations, prompts, and checkpoints were tried before the reported one.

If ten were tried and the best was reported, the number is a maximum, not a measurement — and the inflation cannot be recovered after the fact. The same question applies to which *result* is quoted from a report containing several.

**If the answer is weak:** stop comparing bidders on their self-selected benchmarks. Specify the evaluation yourself, in advance, and require every bidder to run it.

## 4 How many independent runs, and what is the uncertainty?

**Ask for:** trials per item, sample size, and a confidence interval.

A single run of a stochastic system is an observation, not a measurement. And a comparison between two systems without intervals does not establish that one is better — a systematic review of 445 published benchmarks found that only 6% included uncertainty estimates, error bars, or statistical tests when comparing results between models.

> **The single most useful follow-up:** *"If you ran this again next week, how different could the number be?"*

**If the answer is weak:** do not treat a difference between two vendors as real. Require repeated runs before acceptance, and specify the number in the bid.

## 5 What was excluded — and who decided the criterion, and when?

**Ask for:** runs planned, runs executed, runs reported, and every exclusion with its criterion.

The arithmetic should reconcile. Unexplained missing runs are the most common integrity defect we encounter, and their direction is not random — runs disappear more often when they fail. An exclusion criterion invented after seeing results requires the pre-exclusion number to be shown alongside.

**If the answer is weak:** the reported figure is not the measured figure. Ask for the number before exclusions, and use that one until the gap is explained.

## 6 How much help did the system get?

**Ask for:** prompt engineering effort, per-task tuning, scaffolding, tool provision, retry policy, best-of-$n$ selection, and any human involvement.

How hard you try to get a good answer is part of the result. This is now internationally recognised as part of evaluation methodology, which means an undisclosed elicitation regime makes the number incomparable to any other number — including the one your other bidder submitted.

> **Ask directly:** *"Was a human in the loop at any point in producing this figure?"*

**If the answer is weak:** the number describes an expert operator under laboratory effort, not your staff in production. Require the same elicitation regime to be reproducible by your own users.

## 7 Could the system have influenced its own score?

**Ask for:** how scoring was performed, whether the evaluated system could write to anything the grader later read, and whether it had network access during the test.

This is the question people forget, and for agentic systems it is often the one that matters most. Where an agent can modify test artifacts, alter state the scorer depends on, or reach the reference solutions, a "success" may record successful manipulation of the grader rather than successful completion of the task. Documented instances of exactly this failure in widely used agentic coding benchmarks are a matter of public record.

> If the answer is not a clear, evidenced **no**, every number from that evaluation is unreliable in an unknown direction — including the ones that look unremarkable.

**If the answer is weak:** the evaluation environment has to be assessed before its results mean

anything. This is the one weak answer that invalidates rather than discounts.

## 8 Was it tested under our constraints — or in a laboratory?

**Ask for:** the evaluation's network access, tool implementations, permission model, latency budget, rate limits, retrieval sources, data quality, and approval gates — and how each compares to yours.

Evaluation conditions are part of the result. A system that scores well with unrestricted network access, idealised tool stubs, clean documents, and no approval step is not the system that will run inside your security policy. **Measure the difference** rather than hoping it is small; in our experience it is routinely double digits.

**If the answer is weak:** you are holding laboratory capability, not operational capability. Make a production-constrained run a condition of acceptance, before go-live rather than after.

## 9 How does it fail — not how often?

**Ask for:** failure modes with severity, whether you would detect them, and whether you could undo them.

An average conceals catastrophic failure. A 97% accurate system with a 3% severe, silent, irreversible error rate is not the same product as a 97% accurate system with a 3% obvious, recoverable one — and no aggregate metric distinguishes them.

Then ask the question that outranks all of these: **"What failure modes did you not test for?"** An untested mode is an open risk, not a passed test.

**If the answer is weak:** size the worst credible failure yourself and design the human checkpoint around it, rather than around the average case.

## 10 Who ran it, who paid for it, and can anyone else reproduce it?

**Ask for:** the evaluator's identity, their financial relationship to the vendor, who set the scope, whether the evaluator could add tests, who held approval over publication, and whether you can re-run the evaluation yourself.

Vendor self-evaluation is documentation, not independent evidence — and third-party status alone does not fix it. An external evaluator working to a vendor-set scope, on vendor-supplied data, under a vendor publication veto, is not independent in any sense that should reassure you.

> **The decisive test:** *"Can we reproduce this ourselves, on documents you have never seen?"* If the answer is no, ask why — the reason is usually more informative than the benchmark.

**If the answer is weak:** the claim rests on trust rather than evidence. Decide consciously whether the consequence of the decision justifies accepting it on that basis.

---

## If the answers are assurances rather than evidence

That is not an unusual outcome, and it is not necessarily a reason to reject a vendor. It is a reason not to let the number carry the weight of the decision.

Three things you can do before signing:

1. **Build a private evaluation set** from your own documents and never release it. It is the only reliable defence against contamination, and it costs less than the first month of a bad deployment.
2. **Make disclosure a requirement of the bid**, not a request afterwards. Once three vendors have each optimised a different benchmark, no downstream analysis fully recovers the comparison.
3. **Measure under your own constraints before acceptance**, not after go-live.

## The principle underneath all ten

> **The question is not simply whether the AI passed the test. The question is whether the test deserves to be trusted.**

## Where this comes from

These questions are the executive-facing surface of **MQ-EVA**, the Maple Quanta Evaluation Assurance framework: seven evaluation dimensions, a five-grade evidence hierarchy from assertion to adversarially validated, and measures for evaluation confidence, the gap between laboratory and production performance, reproduction agreement, and the maximum credible evaluation failure.

The full methodology, the Evaluation Disclosure Sheet template, machine-readable schemas, and a complete worked example are published openly:

- White paper — *Independent AI Evaluation Assurance: A Framework for Determining Whether AI Performance Claims Can Be Trusted*: maplequanta.ca/resources
- Service and methodology overview: maplequanta.ca/ai-evaluation-assurance

**Maple Quanta Inc.** provides Independent AI Evaluation Assurance — independent assessment of whether the evidence behind an AI claim is strong enough to justify the decision it is supporting.

Ottawa, Canada | maplequanta.ca

*This guide is general information, not advice on a specific procurement. Maple Quanta is not an accredited certification body and does not certify AI systems.*