Autoresearch Playbook

THE 12-POINT ASSESSMENT · 2026

Is your AI prompt production-ready?

Score your prompt across the 12 checks that separate a demo from something you can put in front of a customer. Get a number — and a verdict.

Before you ship that prompt

A prompt that passes five hand-tests can still fail catastrophically on the sixth real input — the empty field, the weird format, the user who pastes "ignore previous instructions." In 2026, with metered API billing, that 1-in-20 failure doesn't just annoy a user; it bills you for the retry and erodes trust at scale. Reliability is now a cost lever. Here's how to measure it.

How to score. Run one real, important prompt through all 12 checks. Score each:

- **0** — not addressed
- **1** — partially handled, or only by luck
- **2** — deliberately engineered and verified

Add them for a total out of **24**. Your verdict is on the last page.

The 12 points

01 Input containment

User text is wrapped in clear delimiters (XML tags, triple quotes), never concatenated raw into your instructions.

The boundary between your logic and someone else's data. Without it, input can smuggle in "ignore previous instructions" and hijack the prompt — the SQL injection of the AI era.

0 raw concat · **1** some delimiting · **2** strict containment

02 Instructions live in the system role

Rules, persona, schema, and policy sit in the system prompt; only the variable task goes in the user turn.

Separation resists overrides and unlocks prompt caching — where the 2026 cost savings come from.

0 one user blob · **1** mixed · **2** clean split

03 Cache-aware ordering

Heavy, unchanging content first; variable content last.

Caching only fires on a byte-identical prefix — and can cut input cost ~90% on repeated calls. One stray leading character breaks it and you pay full price.

0 interleaved · **1** partial · **2** stable-first

04 Reasoning style matches the model

Standard models: show work step by step. Reasoning / extended-thinking models: give the goal and success criteria, not the steps.

The wrong style either fights the model's own process or loses accuracy on multi-step work.

0 none · **1** one-size-fits-all · **2** matched

05 One job per prompt

Multi-step work is a chain of focused prompts, not one mega-prompt.

A prompt juggling five jobs is impossible to debug and overloads attention. Atomic steps fail loudly, in one place.

0 one giant prompt · **1** loosely split · **2** atomic chain

06 An explicit output contract

Specify the exact shape — keys and types, e.g. `{"items":["string"],"count":int}` — with a separate field for the model's thinking.

"Give me a list" yields a different shape every run and breaks the code that consumes it.

0 freeform · **1** loose · **2** typed schema + thinking field

07 Concrete negative constraints

Specific prohibitions — "Do not output markdown fences. Do not add a preamble." — not "don't be wordy."

Models follow explicit "do not X" far better than abstract style requests.

0 none · **1** vague · **2** specific

08 Programmatic validation first

Automated checks confirm the output parses and has every required field before any quality judgment; a structural fail scores zero and retries.

Catch broken output in code, not in front of a customer.

0 eyeballed · **1** spot-checks · **2** automated gate

09 An adversarial test set

A "golden set" of 20–100 cases — ~70% realistic, ~30% nasty (empty, gibberish, injection, long context).

Five happy-path tests never surface the 1-in-20 failure. The adversarial 30% is where production breaks.

0 ad-hoc · **1** happy-path only · **2** realistic + adversarial

10 Layered success metrics

Pass/fail thresholds on structure (does it parse), accuracy (facts right), and style (judged by a model where subjective).

"Looks good" is not a metric. You can't keep the winning variant if you can't measure which one won.

0 vibes · **1** one metric · **2** three layers

11 Versioned, with regression tests

Prompts are versioned like code; every change re-runs the whole golden set before promotion.

Fixing one edge case silently breaks another — the regression trap. And a model update can degrade an untested prompt overnight with no change on your side.

0 no versions · **1** versioned, no re-test · **2** versioned + regression

12 Model fit & a human gate

Temperature matches the task (low for extraction/JSON), instructions are tuned to the model's dialect, and high-stakes outputs route to a human below a confidence threshold.

Full automation on a critical path is how a quiet 5% error becomes a public catastrophe.

0 defaults, full auto · **1** some tuning · **2** tuned + human-in-the-loop

Your score

21–24	**Production-ready.** Ship it — with monitoring and a versioned prompt library so it stays that way.
14–20	**Risky.** It works today and will embarrass you later. Close every 0 and 1 before real traffic.
0–13	**Not ready.** High odds of a quiet, expensive failure. Engineer it before it touches a customer.

Don't score prompts by hand. Make them test themselves.

The Autoresearch Playbook turns these 12 points into an automated loop: an AI agent writes prompt variants, scores each against these criteria, keeps the winner and reverts the rest — overnight, and $0 on a local model for non-training work.

Start free: autoresearchplaybook.com/assessment · The 12 templates ($97): autoresearchplaybook.com

© 2026 Autoresearch Playbook