SUMATOSOFT

25 Questions to Diagnose Your AI Spend

AI Cost Audit Checklist

How to use this checklist

Score yourself in 15 minutes

This checklist diagnoses where AI cost is hiding in your stack. It is not theoretical. Each question maps to a real pattern we see in enterprise AI deployments — patterns that, once addressed, recover real money.

How to use it:

1. Read each question. Answer yes or no honestly. If you are uncertain, the answer is no.
2. Tick the box next to every question you can answer yes. Leave the rest unchecked.
3. Count your unchecked boxes at the end. The score table on the last page tells you where you stand and what to do next.

Time required: ~15 minutes. Recommended: do this with at least one engineer present.

The checklist covers five layers of AI cost discipline:

Section	Layer	What it surfaces
1	Visibility & Observability	Whether you can see your AI spend in time to act on it
2	Model & Inference Choices	Whether you are using the right model for each request
3	Architecture	Whether your system shape is cost-aware
4	Operations & Prompts	Whether your operational hygiene scales with usage
5	Financial Governance	Whether AI spend is owned, predictable, and tied to unit economics

Section 1 of 5: Visibility & Observability

If you cannot see AI cost as it happens, you cannot govern it. Most enterprise AI bills spike between monthly statements — by the time the invoice arrives, the budget is gone.

☐ **Q1. Can you state your current monthly AI spend within $1,000 without checking a billing dashboard?**
Why it matters: If you cannot, you are flying blind. AI cost moves daily, not monthly.
Red flag: Wide ranges, hesitation, or "I'd need to ask DevOps."
If you answered "no": Install Helicone, Langfuse, Vantage, or Portkey within one week. Single config change for most stacks.

☐ **Q2. Are your AI workloads tagged by feature, team, and customer cohort?**
Why it matters: Without tagging, you cannot attribute cost to value — and you cannot defend AI spend to the CFO.
Red flag: "We have aggregate billing only."
If you answered "no": Tagging is a 1-week engineering task with permanent ROI. Make it the first AI FinOps deliverable.

☐ **Q3. Do you have a real-time dashboard showing AI cost trends — not just monthly reports?**
Why it matters: Spikes are sub-monthly. CSV-export workflows are always two weeks stale.
Red flag: Monthly Finance exports, manual spreadsheets, no live view.
If you answered "no": Most observability tools render this out of the box. Set up before the next sprint.

☐ **Q4. Can you tie AI cost to a specific business outcome (revenue, support deflection, retention)?**
Why it matters: If you cannot, leadership cannot justify continued investment — and you cannot ask for more.
Red flag: "AI is a cost center, not a metric."
If you answered "no": Define one measurable outcome per AI feature this quarter.

☐ **Q5. Do you set per-feature monthly cost budgets with automated anomaly alerts?**
Why it matters: Cost spikes are predictable; alerts are cheap. Manual review is too late.
Red flag: "We notice when the bill arrives."
If you answered "no": Configure threshold alerts in your observability tool. Should take an afternoon.

Section 2 of 5: Model & Inference Choices

Model selection is the highest-leverage cost decision in AI. Most teams default to a frontier model at launch and never re-evaluate. The cost gap between frontier and small models is 10–30×.

☐ **Q6. Do you know what percentage of your requests use a frontier model (GPT-4-class, Claude Opus) vs. smaller models?**
Why it matters: Frontier models cost 10–30× more per token. If you do not know the split, you are overpaying — often dramatically.
Red flag: "We default to the big model for everything."
If you answered "no": Add model-tier breakdown to your observability dashboard this week.

☐ **Q7. Have you tested whether smaller / cheaper models match your quality requirements for at least 50% of request types?**
Why it matters: Most teams never benchmarked alternatives. Models improve quarterly; what required frontier 6 months ago often does not now.
Red flag: "We chose our model at launch and never re-evaluated."
If you answered "no": Run a weekend evaluation on 100 sample requests with a small model. Typically reveals 40–70% routable to cheaper tier.

☐ **Q8. Are you using prompt caching for system prompts that repeat across requests?**
Why it matters: Anthropic and OpenAI both support native prompt caching with 50–90% savings on cached tokens. Most teams have not turned it on.
Red flag: "We don't know what prompt caching is" or "we never enabled it."
If you answered "no": Single config change in your API client. Highest-ROI hour you will spend this quarter.

☐ **Q9. Do you have any AI workloads that could tolerate latency above one hour but currently run synchronously?**

Why it matters: Batch APIs offer a 50% discount for non-urgent work. Most "real-time" workloads are not actually time-sensitive.

Red flag: Document processing, report generation, scheduled enrichment running through real-time endpoints.

If you answered "no": Migrate to Anthropic Batch API or OpenAI Batch API. Immediate 50% reduction on those workloads.

☐ **Q10. When did you last benchmark a smaller model against your current production one for your specific tasks?**

Why it matters: Model capability moves quarterly. Yearly re-evaluation is the minimum baseline; quarterly is best practice.

Red flag: "Never" or "at launch."

If you answered "no": Add a quarterly model-benchmark cadence to your engineering calendar.

Section 3 of 5: Architecture

☐ **Q11. Do you know your average input token count per request — and whether it is trending up over time?**
Why it matters: Input tokens are typically the dominant cost lever. Silent input growth is the #1 cause of mystery cost spikes.
Red flag: "We don't track input length."
If you answered "no": Add input-length metrics to your observability layer this sprint.

☐ **Q12. Do you use RAG to retrieve only relevant context, or do you stuff full documents into the prompt?**
Why it matters: 100K-token context windows are cost traps when you only need 2K of relevant content. RAG can cut input costs 80–95%.
Red flag: "We rely on the long context window" or "we send the whole doc just in case."
If you answered "no": Evaluate a RAG architecture for your document-heavy workloads. Typical ROI: payback in weeks.

☐ **Q13. If you fine-tuned a model, did you formally evaluate RAG as a cheaper alternative first?**
Why it matters: Fine-tuning has high upfront and maintenance cost. RAG often achieves equivalent quality at a fraction of the spend.
Red flag: "We fine-tuned because it was the obvious choice."
If you answered "no": Re-evaluate at next major model release. New base-model capability often makes fine-tuning unnecessary.

☐ **Q14. Do you cache semantically similar queries, or recompute every request from scratch?**
Why it matters: Customer support, FAQ-style, and search workloads have huge repeat rates — often 40–70% within an hour.
Red flag: Same questions answered hundreds of times per day with zero caching.
If you answered "no": Add semantic caching (Redis + embeddings, or managed via Portkey / GPTCache). Quick win with 30–60% savings.

☐ **Q15. Could a cheaper classical ML model (logistic regression, small BERT) handle the simple cases before they reach the LLM?**
Why it matters: Hybrid architectures route only complex cases to expensive models. Pre-filtering catches 50–80% of requests for near-zero cost.
Red flag: "Everything hits the LLM."
If you answered "no": Add a pre-filter classifier for known intents. Typical implementation: 2–4 engineer-weeks.

Section 4 of 5: Operations & Prompts

Prompt engineering debt accumulates silently. System prompts grow, instructions get added for edge cases that no longer apply, and no one ever audits them. 2K-token prompts × millions of requests = real money.

☐ **Q16. When did you last audit your production system prompts for length and necessity?**
Why it matters: Prompts accumulate instructions over time, many of which apply to edge cases that no longer exist. 30–50% size reduction is typical with no quality loss.
Red flag: "At launch" or "never."
If you answered "no": Schedule a quarterly prompt audit. Add it to your engineering planning cadence.

☐ **Q17. Do your AI responses have max_tokens caps and structured output formats (JSON mode, function calling) where applicable?**
Why it matters: Unbounded output equals unbounded cost. Verbose freeform responses are expensive by default.
Red flag: "We let the model decide length."
If you answered "no": Add max_tokens to every API call. Convert free-form responses to JSON / structured output where possible.

☐ **Q18. Are your prompts version-controlled with regression tests for cost-per-response, not just quality?**
Why it matters: Quality regression tests are common; cost regression tests are rare. Both matter equally.
Red flag: "We don't track prompt cost over time."
If you answered "no": Add a cost metric to your prompt CI alongside quality scores. Use PromptLayer, Helicone evaluations, or Braintrust.

☐ **Q19. Do your prompts include explicit instructions to keep outputs concise where appropriate?**

Why it matters: "Respond in under 100 words" reduces output tokens 40–60% for many use cases at minimal quality impact.

Red flag: Free-form prompts with no length guidance.

If you answered "no": Audit prompts for length-control instructions. Add explicit conciseness directives where appropriate.

☐ **Q20. Have you measured whether expanding context actually improves output quality for your use cases — rather than assumed it does?**

Why it matters: Teams add context "to be safe." Often it costs more without improving outputs. Verify, don't assume.

Red flag: Context size grew over time without re-evaluation.

If you answered "no": A/B test reduced context vs. current setup on a sample of representative queries.

Section 5 of 5: Financial Governance

☐ **Q21. Is there a single owner accountable for AI cost variance month-over-month?**
Why it matters: Without an owner, no one optimizes. Diffuse responsibility produces diffuse results.
Red flag: "Engineering and Finance both kind of look at it."
If you answered "no": Name an AI FinOps lead with explicit responsibility for cost outcomes. Add to next quarter's org plan.

☐ **Q22. Do engineers see real-time cost impact of their architecture choices in their workflows?**
Why it matters: Engineers optimize for what they see. If cost lives in a dashboard nobody reads, nobody optimizes.
Red flag: Cost data is in a Finance system disconnected from engineering tooling.
If you answered "no": Surface cost-per-request in development environments. Tools like Helicone and Portkey support this natively.

☐ **Q23. Do you have a quarterly review process for prompt and model optimization?**
Why it matters: AI cost decay is slow. Quarterly cadence catches drift before it becomes a budget crisis.
Red flag: "We optimize when the bill spikes."
If you answered "no": Add prompt + model review to your existing OKR or planning cadence. No new process needed.

☐ **Q24. Is AI cost included in your unit economics (per customer, per feature, per transaction)?**
Why it matters: AI features often appear profitable in aggregate but lose money on heavy users. Per-customer cost is the only honest view.
Red flag: "We look at total AI spend only."
If you answered "no": Build per-customer cost attribution. Typical finding: 20% of users consume 80% of AI cost; pricing may not reflect this.

☐ **Q25. Can you predict next quarter's AI spend within ±15% based on planned product changes?**
Why it matters: Predictability is the ultimate FinOps maturity metric. Without it, AI cost remains a budget risk in every cycle.
Red flag: "We hope the bill doesn't blow up."
If you answered "no": Build a simple spend forecast based on user growth + planned feature rollouts. Iterate quarterly until ±15% is reliable.

Your result: Score yourself

Count the number of questions you could NOT answer "yes" to. The total — out of 25 — places you in one of four bands.

Score	Band	What it means
0–5	Solid foundation	Your AI cost discipline is mature. Focus on unit-economics refinement and edge-case optimization. Re-take this checklist next quarter.
6–12	Material opportunity	Typical for mid-market teams 12–18 months into AI. Recoverable spend: 30–50% within 90 days using the right mechanisms. Quick wins in Sections 1 and 4.
13–18	Significant savings available	AI cost is running ahead of architecture. Recoverable spend: 50–70% within 90 days. Start with observability (Section 1), then architecture (Section 3).
19–25	Critical — architectural intervention needed	AI cost is likely a budget risk in your next cycle. Recoverable spend often exceeds 70%. Begin immediately with Section 1, and consider an external audit before the next budget review.

Reading your score by section

If your unchecked boxes cluster in one section, that's your starting layer:

Most unchecked in Section 1: start with observability — you cannot fix what you cannot see.

Most unchecked in Section 2: tiered routing and prompt caching will yield the fastest wins.

Most unchecked in Section 3: architectural changes (RAG, semantic caching, hybrid models) — larger effort, larger payoff.

Most unchecked in Section 4: schedule a prompt audit this month; the ROI is fastest and effort is lowest.

Most unchecked in Section 5: this is an organizational problem, not a technical one. Name an owner first.

SUMATOSOFT

Thank you for your time!

Any questions? Drop us a line!

Headquarters

One Boston Place, Suite 2602
Boston, MA 02108, United States

Other ways to get in touch

info@sumatosoft.com
sumatosoft.com