LOTUSNEX

Production AI Readiness Checklist

12 architectural questions your AI system should answer before it ships to real users.

Most AI systems don't fail because the model is wrong. They fail because the surrounding system was never engineered for production — no evaluation harness, no defined failure modes, no observability, no clear ownership after handoff.

This checklist surfaces the gaps most teams discover too late: after the demo, after the launch, after the first incident. Work through it before you ship. If you can answer every question with confidence, your system is production-ready. If you can't, you know exactly where to focus.

How to use this document: Read each question. If you can answer it concretely — with a specific system, document, or process — check it off. If the answer is vague or 'we'll handle it later,' mark it as a gap. A pattern of gaps in any single section indicates a systemic risk.

Section	Questions	Risk area
01 Failure modes	1–3	Silent degradation, undefined behavior
02 Evaluation	4–6	Quality drift, regression, hallucination
03 Security & access	7–9	Data exposure, unauthorized access
04 Observability & cost	10–11	Runaway spend, invisible failures
05 Ownership & handoff	12	Operational debt, team dependency

SECTION 01

Failure modes

Production systems fail. The question is whether yours fails safely. These questions test whether your failure behavior is designed, not accidental.

01

Can your system fail deterministically — or does it degrade silently?

Silent failures (a model returning a plausible but wrong answer, a retrieval pipeline returning stale results) are harder to detect and more damaging than explicit errors. Every failure mode should produce a defined, observable output.

02

Have you defined what the system must never do — and verified it can't?

Guardrails are not a prompt engineering problem. They require explicit boundaries, test cases covering adversarial inputs, and automated checks that run in CI. If 'don't do X' only lives in the system prompt, it is not a guardrail.

03

Do you have a human-in-the-loop path for actions above a defined risk threshold?

Fully autonomous AI actions are appropriate for low-stakes, high-volume operations. Anything that modifies external state, sends communications, or has financial or compliance implications should have a human approval gate with a clear escalation path.

SECTION 02

Evaluation

If you can't measure whether your system is working, you can't know when it stops working. Evaluation is not QA — it's structural.

04

Do you have an offline evaluation harness with a fixed test set?

A test set is a collection of inputs with known correct outputs that you run against the system before every release. Without this, you have no baseline and no way to detect regression. 'It seemed fine in testing' is not an evaluation strategy.

05

Can you detect quality drift between deployments?

Model providers update underlying models. Prompts that worked in January may behave differently in March. Your evaluation harness should run on every deployment and alert when output quality drops below a defined threshold.

06

Have you measured retrieval quality — not just retrieval speed?

Most RAG implementations are tuned for latency, not accuracy. You should have precision and recall baselines for your retrieval pipeline, and regression checks that run when the index is updated. 'It returns results' is not a quality metric.

SECTION 03

Security and access control

AI systems that touch real data carry real access control requirements. These are not features you add later — they are architectural decisions that must be made before you build the retrieval or agent layer.

07

Is access control enforced at retrieval time — or only applied after the fact?

Post-filtering (retrieving everything and then hiding results) is not secure. Under certain query patterns, post-filtered systems expose content to users who should not have access. Permission enforcement must happen at the index query layer, not downstream.

08

Do you have an audit trail for every action the system takes on behalf of a user?

For any AI system operating in a regulated industry or handling sensitive data, audit logs are a compliance requirement, not a nice-to-have. Logs should capture: who made the request, what data was accessed, what action was taken, and what the outcome was.

09

Has your tool permission model been reviewed for least-privilege access?

Agent systems with broad tool access are a significant security surface. Each tool integration should have the minimum permissions required for its function. If your agent can read and write to a system, ask whether it needs write access at all — and whether that access should require human approval.

SECTION 04

Observability and cost

You cannot operate what you cannot see. These questions test whether you have the instrumentation to know what your system is doing, what it costs, and when something goes wrong.

10

Do you have cost telemetry — per request, per user, or per workflow?

Token costs compound at scale in ways that are not intuitive. A system that costs $0.02 per request may be acceptable at 100 requests per day and catastrophic at 10,000. You need per-request cost visibility, token budgets, and alerting before you hit a threshold — not after.

11

Do you have traces and logs for every tool call and model invocation?

When something goes wrong in a multi-step AI system, you need to reconstruct exactly what happened: which tools were called, in what order, with what inputs and outputs, and where the chain broke. Without trace-level logging, incident response is guesswork.

SECTION 05

Ownership and handoff

A system that only the people who built it can operate is not a production system — it's a prototype with users. This final question tests whether your system is built to be owned.

12

Could your team operate, debug, and extend this system without the people who built it?

This is the handoff test. It requires: runbooks covering the most common failure scenarios, documentation that reflects actual system behavior (not intended behavior), clear interface contracts between components, and an on-call process for when the system behaves unexpectedly at 2am. If the answer is no, the system is not ready for production — regardless of how well it performs in a demo.

HOW TO INTERPRET YOUR RESULTS

Score	What it means	Recommended next step
10–12 checked	System is production-ready in these dimensions.	Monitor and maintain. Document baselines.
7–9 checked	Significant gaps present. Manageable with focused effort.	Prioritize by section. Address evaluation and access control first.
4–6 checked	System is not production-ready. High operational risk.	Architecture review before any further feature work.
0–3 checked	Prototype-stage architecture with production traffic.	Stop adding features. Rebuild the production layer.

LOTUSNEX

Found gaps you're not sure how to close?

We run Architecture Reviews for seed-to-Series A B2B SaaS companies building on AI. In 30 minutes, we'll map your current system against production requirements, identify your highest-risk gaps, and give you a concrete path forward — whether you work with us or not.