Xtensyon.
TECHNICAL BRIEF

Building Evaluation Sets from Real Tickets: A Practical Alternative to Benchmarks

November 6, 2025 • Xtensyon Labs • 7 min read

> *Public benchmarks rarely match internal workflows. This brief shows how to turn a set of real tickets into a repeatable evaluation suite that catches regressions after prompt or indexing changes.*

TL;DR

- Sample tickets by workflow, not by volume, so the suite stays representative.
- Store expected sources and pass criteria, not full expected answers.
- Run the suite on every release and keep results per build.
- Add edge cases from incidents right away so they do not return.

Executive Summary

Evaluation is easier when it starts from real work. We describe how to build a ticket-based evaluation set that teams can run on every release. The method focuses on coverage, not academic scoring: select representative workflows, define what “pass” looks like, and track results over time. The key is keeping the set small enough to run often and strict enough to catch meaningful regressions.

Why It Matters

Teams lose confidence when assistants change behavior unexpectedly. Small indexing tweaks can break citations. Prompt changes can fix one workflow and quietly damage another. A ticket-based evaluation set makes changes safer and speeds up iteration because failures are concrete and tied to business tasks.

What We Built

- A sampling method that selects tickets across workflows, complexity, and required permissions.
- A pass criteria template covering source correctness, required entities, and safe refusal cases.
- A replay harness that logs retrieval context and compares results per release.
- A process for adding new tests from incidents and supervisor feedback.

Observed Outcomes

- Fewer regressions shipped after gating prompt and indexing changes on the suite.
- Quicker root-cause analysis because failing cases carried retrieval and metadata traces.
- Improved stakeholder trust once quality was reported as trends, not anecdotes.

Implementation Notes

- Redact sensitive ticket data before storing it in test fixtures.
- Keep expected sources flexible. Focus on correct policy or runbook, not exact wording.
- Tag each test by workflow so teams can see which areas are affected by a change.
- If the suite grows too big, rotate: keep a core set and a weekly extended set.

Governance & Risk

- Test data is still data. Apply retention, access, and redaction rules.
- Avoid using customer identifiers in prompts or fixtures.
- Document who can modify the suite; quality gates should be controlled.

Release Checklist

- Do tests cover key workflows and not only frequent tickets?
- Do pass criteria include citation or source checks?
- Is the suite run on every release?
- Are results stored and comparable over time?

- Do incidents feed back into new tests quickly?

Conclusion

A good evaluation set looks like your work, not a benchmark leaderboard. Start from tickets, define pass criteria, and run the suite often. Teams will ship changes with fewer surprises.

Keywords

evaluation | llm testing | support tickets | quality | rag | regression