

Feedback Guide

This guide covers what feedback to send, how to A/B test so your feedback includes evidence, and where to send it.

What we want to hear

- **Wrong or incomplete answers.** The query you ran, what came back, and what should have come back. This is the highest-value feedback there is.
- **Surprising wins.** An answer that saved real time, found something nobody knew, or changed a decision.
- **Indexing gaps.** Code, projects, or relationships you expected in the graph that aren't there.
- **Friction.** Setup problems, auth failures, slow queries, confusing errors.
- **Missing capabilities.** Questions your team wanted to ask that the graph can't answer yet.

How to A/B test

The most persuasive feedback compares the same question with and without Orbit, in whichever surface your team uses: Duo and/or an external agent like Claude Code or Codex. A good A/B captures four things per run: wall time, tool calls, approximate tokens consumed, and the answer itself, followed by an assessment of which answer you would actually act on.

The Orbit A/B Test skill does this for you

Claude Code: run this one command in your repo, then invoke `/orbit-ab-test "your question"` :

```
mkdir -p .claude/skills/orbit-ab-test && curl -fsSL -o .claude/skills/orbit-ab-test/SKILL.md https://orbit-dp-program-3fc9a6.gitlab.io/resources/orbit-ab-test-skill.md
```

Other agents: download the [skill file](#) and paste its contents as instructions with your question. The skill runs your question twice in isolated contexts, once with Orbit and once without, measures both runs, and produces a structured report: a comparison table, the agent's own scoring on completeness and groundedness, what one run found that the other missed, and a human verdict line for you to fill in.

The agent's self-scores are approximate. The line we weigh most is your **human verdict**: which answer would your engineer act on? Fill it in every time.

Where feedback goes

1. **The Orbit feedback epic** (public): gitlab.com/groups/gitlab-org/orbit/-/work_items/4. Paste A/B reports and findings as comments. We respond to every thread within one business day, and bugs are triaged as high priority during the program.
2. **Your weekly sync** for anything nuanced, strategic, or that needs a conversation.
3. **Your program contact directly** for anything urgent or sensitive.

Before posting to the public epic, redact. The epic is world-readable. Remove private file paths, class and function names, repository names, and any code. Keep the shape of the finding ("a caller in a second repo was missed") and strip the identifying detail. If a finding can't be shared without exposing your code, bring it to the weekly sync instead.

Weekly routine

- Run at least one A/B per week against one of your three committed use cases.
- Post the report the same day, while the context is fresh.
- Bring the most surprising result, good or bad, to your weekly sync.