How to Verify That Spec Prioritization Is Working

Verification depends on how you implemented prioritization. These are the standard, clean checkpoints used in real projects.

---

1. Verification when using a dedicated “critical tests” job

a) Check GitHub Actions

- You will see two jobs in the workflow:
    - `critical-tests`
    - `full-regression`
- The **critical tests job starts first**.
- `full-regression` begins **only after** `critical-tests` completes (because of `needs:`).
- Total run timeline clearly shows:
    - Critical suite → completed first
    - Regression suite → starts later

This confirms that your pipeline executes critical tests with higher priority.

b) Check Cypress Cloud

- There will be a **separate test run** for the critical suite.
- The timestamp of the critical run is earlier than the regression run.
- Only the critical specs appear in that earlier run.

This validates both order and scope.

---

2. Verification when prioritization is based on folders or tags

a) Check the Cypress command in CI logs

The first Cypress run should include:

- Spec filtering, for example:
    - `--spec cypress/e2e/critical/**`
- OR tag filtering, for example:
    - `--env grepTags=critical`

The logs show which spec files were picked up and executed first.

b) Check test list in Cypress Cloud

- The first run shows only critical-tagged or critical-folder specs.
- The second run shows all remaining specs.

The execution order in Cloud confirms prioritization.

---

3. Verification through runtime behavior

A simple confirmation method:

- The **critical suite finishes quickly** and produces feedback early.
- The full suite (regression) runs later and takes longer.

This is the practical effect of prioritization—fast signal, slower work later.