

Flakiness issue has caused a large portion of the [Flutter tree](#) redness, and below workflow will be enforced to reduce flaky issues. The framework post-submit DeviceLab tests will be focused on in the beginning, and the logic will be extended to other host only tests in the future.

From [Flutter tree dashboard](#), a flake is identified as a box with an exclamation icon. There are two types that will result in same flaky box.

- Multiple reruns on the same commit and same task (earlier run fails, but the last run succeeds). For this case, check logs by clicking different build runs.

[[/images/task_flake_multiple_builds.png|width=300px]]

- A single run on the same commit and same task, but multiple reruns from test runner. For this case, check logs by clicking `stdout` of the test step: it shows data about failed or succeeded runs in the end ([example](#)). See <https://github.com/flutter/flutter/wiki/Understanding-a-LUCI-build-failure> for how to locate the test step and `stdout`.

[[/images/task_flake_test_runner.png|width=300px]]

Preventing flaky tests

Adding a new DeviceLab test

DeviceLab tests are located under [/dev/devicelab/bin/tasks](#). If you plan to add a new DeviceLab test, please follow

- Create a PR to add test files
 - Make sure an ownership entry is created for the test in [TESTOWNERS](#) file
- Enable the test in the staging env. first for validation - make sure 50 consecutive successful runs without any flakiness issue
 - How: add the new test to the appropriate platform in [devicelab_staging_config.star](#)
 - Monitor the test execution in the [milo dashboard](#)
- If no flakiness issue pops up, then enable the test in the prod env. and you will see the new test in the [build dashboard](#).
 - How
 - Add the new test to the appropriate platform in [devicelab_config.star](#)
 - Enable the test to [build dashboard](#) by adding an entry in [.ci.yaml](#)

Detecting flaky tests

On a weekly basis, [an automation script](#) will scan through test execution statistics over the past 15 days and identify top flaky ones

- If there are any test builders whose Flaky Ratio $\geq 2\%$
 - Create a tracking bug if not existing in the [bug.pool](#).
 - The sub-team TL will be assigned by default for further triage/re-assign.
 - P1 will be labeled
 - If it is not a shard test, the script marks the tests as flaky by updating the entry in [.ci.yaml](#).
 - Add a `# TODO(username): github issue url` above the `bringup: true` line

If an issue is closed, there will be a grace period of 15 days before the automation script refile the issue if the same flakiness persists.

Fixing flaky tests

The TL will help triage, reassign, and attempt to fix the flakiness.

If fixed, the test can be re-enabled after being validated for 50 consecutive runs without flakiness issues (task without exclamation point in flutter build dashboard and task not failed due to the same flaky failure). This can be done by updating the test entry in [.ci.yaml](#).

If not fixable, the test will be removed from the flutter build dashboard or deleted from CI completely depending on specific cases.