



Flutter

Creating a Google testing queue

SUMMARY

Creating better transparency on blocked changes, and tracking for resolution

Author: Kate Lovett (@Piinks)

Go Link: flutter.dev/go/goog-testing-queue

Created: 02/2025 / **Last updated:** 04/2025

WHAT PROBLEM IS THIS SOLVING?

A screenshot of a Google Testing queue status interface. It features a dark background with several sections. The first section, 'Changes approved', has a green checkmark icon and states '1 approving review by reviewers with write access.' Below this, there are links for '1 approval' and '2 pending reviews'. The second section, 'Some checks were not successful', has a green circle with a red dot icon and states '1 failing, 102 successful checks'. Below this, there is a link for '1 failing check'. The third section, 'Google testing', has a red 'x' icon and states 'Failing after 56m — Google Testing', with a link for '102 successful checks'. The final section, 'No conflicts with base branch', has a green checkmark icon and states 'Merging can be performed automatically.'

It is fairly common for framework related pull requests in flutter/flutter to be blocked on the Google testing shard (failing tests from inside Google), which requires a Googler to intervene to unblock the change for contributors.

In this doc, the term contributor is used as a convenient shorthand to refer to pull request authors that do not work for Google, as Googlers typically own the changes they have authored to manage any blockers to land in Google. This is a taxonomy just for this doc, as Googler or not, all Flutter contributors are contributors. 😊

After investigating failures from this presubmit test shard, intervention from a Googler to unblock the pull request is required, and entails:

- Providing a publicly shareable reproduction of the failing Google test so the contributor can modify their change to make it non-breaking
 - Or a description of the unexpected failure from something like a screenshot test.
- And/or working with the Google customer to update client code so that the change can land without breaking anything
 - Typically a fix to client code through the tooling presubmit tooling so that the change to client code in Google can land with the roll of the change from Github.
 - In this case, the change should be treated as a breaking change and follow our [policy](#).
- And/or if the change is not possible to be made non-breaking, and is not a break we want to introduce, the Googler informs the contributor that we cannot accept the change.
 - This is still after investigating to understand the issues, and collaborating with the contributor to try to find resolution.
- And/or identifying if the test failures are actually false negatives, and resolve to unblock the change.
 - These could be flakes, failures from the tests being already broken at that moment in Google, or a separate PR being grouped with the PR in question by our tooling that recreates Github changes in Google for the purposes of this testing.

A given pull request can trigger Google failures from more than one customer, sometimes producing hundreds or thousands of failures.

The Google testing shard is only triggered once a pull request has been approved, unless the author is a Googler themselves. Often, these failures appear when a change is ready to land.

While writing this on Feb 5, 2025, there were [24](#) open and approved framework pull requests in flutter/flutter, and [9](#) were blocked on Google testing.

When pull requests blocked by Google test failures pile up, the Google team becomes a bottleneck for resolving blockers for the changes to move forward. This can be time consuming work, digging through client code, learning customer tests to understand expectations, recreating internal code for safely sharing on Github, etc. Meanwhile, contributors don't have a lot of visibility to what is happening while they wait.

Typically the reviewer of a given pull request will take ownership of a failing Google test. Alternatively, if the author is a Googler, the author takes up resolving it. This can become

problematic when a Googler finds themselves (through coincidence) with multiple pull requests failing Google testing that all need to be resolved.

The experience for contributors is a significant pain point, having no visibility to these failures, no understanding of how many we might be working through at a given time, or the status of their pull request.

Goals

- Create visibility around what pull requests are currently blocked on Google testing failures.
- Create visibility around who is working on resolving them.
- Create visibility around who might be getting swamped with multiple blocked pull requests.
 - Load balance when this happens.
- Restructure unblocking these changes in queue fashion rather than concurrently.
 - Create visibility on which ones are actively being worked on to resolve
 - Prioritize pull requests fixing urgent issues
- Communicate clear expectations to contributors when their pull requests are blocked on this testing shard.

Creating a Google Testing Queue

To resolve some of these pain points, and improve the contributor experience, I'd like to propose a new process around pull requests that are blocked on Google testing.

This process will center around a Github project to track pull requests that are blocked on Google testing, who is working to resolve them, and where they are in the queue in relation to other blocked changes.

This would allow any Googler or contributor to at a glance see what is currently blocked on Google testing:

Google Testing Queue

All blocked PRs By owner + New view

is:open

Title	Assignees
1 Fix Status Bar Icon Brightness #162297	
2 [beta] Show error on macOS if missing Local Network permissions #162119	
3 Fix DropdownMenu default width does not take label into account #161219	QuncCccccc
4 Add a isSystemTextScaler matcher #160120	LongCatIsLooong
5 fix: Dispose codec after completing frame creation #159945	
6 Exclude focus for child when it is hidden in Visibility if maintainInteractivity is not... #159133	Piinks
7 fix: time selector separator in timepicker is not centered vertically #157402	MitchellGoodwin
8 Change cardTheme, dialogTheme, and tabBarTheme type to xxxThemeData #157292	QuncCccccc

We can also see who is working to resolve blockers, and load balance as needed:

Google Testing Queue

All blocked PRs By owner + New view

Filter by keyword or by field

Title	Assignees
LongCatIsLooong 1	
1 Add a isSystemTextScaler matcher #160120	LongCatIsLooong
+ Add item	
MitchellGoodwin 1	
2 fix: time selector separator in timepicker is not centered vertically #157402	MitchellGoodwin
+ Add item	
Piinks 1	
3 Exclude focus for child when it is hidden in Visibility if maintainInteractivity is not ... #159133	Piinks

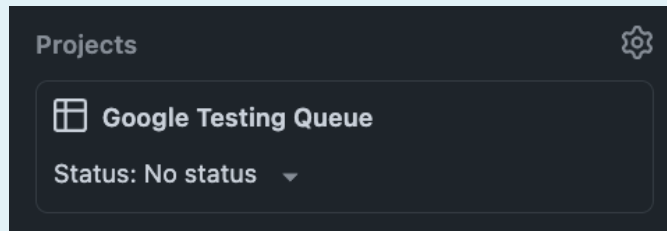
This would include a column in both views that can show which one, per person, they are actively working to resolve.

This will provide transparency to contributors wondering when their PR will be able to move forward.

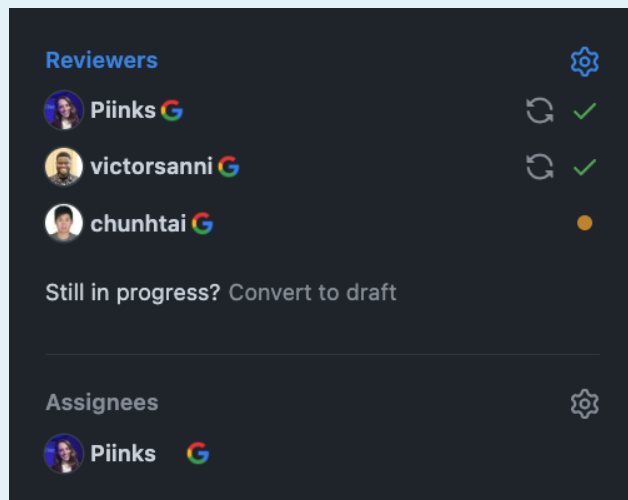
With this in place, ideally the workflow would be as follows

1. Normal code review!
2. When a Google testing failure occurs in presubmit testing, the Googler reviewing or authoring the change confirms it is a real failure that needs to be investigated, and adds it to the queue project.

- a. In the course of regular triage meetings (framework, text input, accessibility, etc), this can be spot checked, adding blocked PRs to the project.



- b.
- 3. The Googler working to unblock the PR is assigned to it.
 - a. This is different from being assigned as a reviewer.
 - i. 'Unblock' here could mean determining it is not a change we want to make, because of its breaking nature. It could also mean working with the contributor to make it non-breaking, or updating client code in Google to allow the change to proceed (this last option usually involves a migration guide on the part of the author in following our breaking changes [policy](#))



- b.
- 4. In the project, a given owner is only expected to be working to resolve one blocked PR at a time. This can be notated as 'in progress' in the project.
 - a. This will accelerate resolution by focusing on one change at a time.
 - b. This is different from how we have been handling these, although we have been working on this informally up until now, we should publicize and document this clearly for contributors.
- 5. Once the change is unblocked [lands/is closed], it will automatically be removed from the project views.
- 6. Add to the list of action items for weekly critical issue triage, which checks for stale or abandoned PRs and ensures high priority issues are making progress, the triage lead can check in on the Google Testing Queue.
 - a. Check for the need to load balance if someone is overloaded with PRs.
 - b. Ensure each person owning a blocked PR has one in progress.
 - c. See if any PRs of high priority need to move to the front of someone's queue.

Documentation

To document this new workflow, we can update this page of our docs to include details of how the new Google Testing Queue will work

<https://github.com/flutter/flutter/blob/master/docs/infra/Understanding-Google-Testing.md>

- This page is woefully out of date, will also bring up to date in general - <https://github.com/flutter/flutter/issues/162832>

It will detail the process above so contributors know what to expect and how to check in on the status of their change.

Also to update, the critical issue triage guide for added instructions to check in on the queue <https://github.com/flutter/flutter/blob/master/docs/triage/README.md#critical-triage>

Announce this via the normal channels - internal, Discord, and flutter-announce.