

# Feature testing

**Note:** The information below is also available as a Wiki page [here](#).

## Purpose of this document

### What this document is

This document outlines the major milestones associated with the quality management of Firefox features. These milestones are centered around manual testing, evaluation and reporting.



The process described below applies only for the Firefox features shipped based on the train model.

### What this document is not

The information available in this document does not apply to:

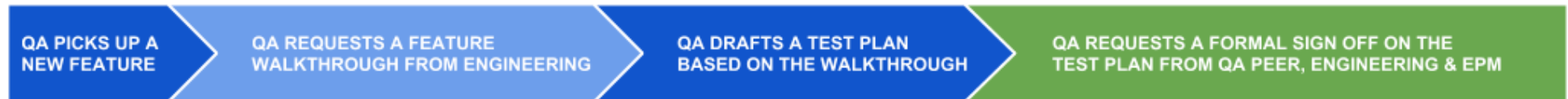
- QA teams conducting automated testing
- QA teams working on off-the-train features



The process described below does not directly result in a final Go/NoGo decision for a Firefox feature.

# Process overview

## Phase 1: Kickoff and Test Plan sign off



- QA requests a feature walkthrough from the engineering team responsible with the feature
  - **Description:** the walkthrough can consist of an actual demo held by the engineering team, a discussion based on a very specific agenda or an email thread tackling a very specific list of topics/questions essential for sketching a Test Plan
  - **Goal:** for QA to understand the scope of the feature and see the big picture
- QA drafts a Test Plan based on the feature walkthrough provided by engineering
  - **When:** immediately after the feature walkthrough
  - **Exceptions:** this might be impossible if the feature is received late and/or test execution is more urgent than Test Plan creation
  - **Best practice:**
    - while not mandatory, it's usually best to consult with the Team Lead if specific items from the Test Plan are unclear
    - once a draft is ready for the Test Plan, an internal review is started by QA peers and/or Team Lead
  - **References:** <https://goo.gl/HdgWft> (Test Plan template)
- QA sends the draft Test Plan to Engineering, Product and QA Manager for review and formal sign off
  - **When:** as soon as the draft Test Plan has been green-lit internally, by QA peer requis and/or Team Lead

## Phase 2: Test preparation and scheduling



- If the draft Test Plan has been signed off by the required parties, QA starts creating high-level test cases based on the Test Objectives described in the Test Plan

- **When:** immediately after the Test Plan has been formally signed off by required parties
  - **Exceptions:** Test Suite drafting could also start while the Test Plan is being reviewed, if QA is confident about the Test Objectives proposed in the Test Plan
- QA sends the high-level Test Cases or the draft Test Suite to Engineering for review
  - **When:** as soon as the high-level test cases have been green-lit internally, by QA peers or Team Lead
  - **Best practice:** it's usually best to send the high-level test cases in the form of a Google Spreadsheet, because not all engineering teams have access to TestRail and it's also a better environment for adding notes and comments per test case or step
- QA starts drafting a Test Suite based on the high-level test cases green-lit by Engineering
  - **When:** as soon as the high-level test cases have been green-lit by Engineering
- QA schedules a testing session with specific activities based on the available time frame
  - **When:** based on priority, time constraints and available bandwidth

## Phase 3: Mid Nightly test execution and sign off



- QA sends out a preliminary status report and schedule update before the mid-Nightly feature sign off
  - **Goal:** For QA to discuss the status of the feature with the Engineering team, prior to the formal sign off.
  - **When:** 1 week before the mid-Nightly sign off
  - **Exceptions:** this might often be impossible if the feature is received late or if testing starts late
- QA formally signs off the feature mid-Nightly
  - **When:** by end of week #3 (6-week or 7-week release cycle) or by end of week #4 (8-week cycle)
  - **Best practice:** while not mandatory, it's usually best to consult with the Team Lead if the status or arguments used in the feature sign off report are unclear or might be subject to interpretation
  - **References:** <https://goo.gl/AEY99U> (sign off template)

## Phase 4: Pre Beta test execution and sign off

QA SENDS A PRELIMINARY STATUS REPORT TO  
ENGINEERING 1 WEEK BEFORE THE PRE BETA SIGN OFF

QA FORMALLY SIGNS OFF  
THE FEATURE PRE BETA

- QA sends out a preliminary status report and schedule update 1 week before the pre-Beta sign-off
  - **Goal:** For QA to discuss the status of the feature with the Engineering team, prior to the formal sign off.
  - **When:** 1 week before the pre-Beta sign off
- QA formally signs off the feature pre-Beta
  - **When:** by end of week #5 (6-week), week #6 (7-week cycle) or by end of week #7 (8-week cycle) -- or 1 week before merge day
  - **References:** <https://goo.gl/AEY99U> (sign off template)

## Phase 5: Pre Release test execution and sign off

QA SENDS A PRELIMINARY STATUS REPORT TO  
ENGINEERING 1 WEEK BEFORE THE PRE RELEASE SIGN OFF

QA FORMALLY SIGNS OFF THE  
FEATURE PRE RELEASE

- (not always applicable) Engineering QA hands over the feature to Release QA and informs the engineering team about it
  - **When:** in the 1st week of the release cycle
  - **Description:** this step is optional, as not all features are handed over
  - **References:** <https://goo.gl/ehzXnK> (feature handover template)
- QA sends out a preliminary status report and schedule update 1 week before the pre-Release feature sign off
  - **Goal:** For QA to discuss the status of the feature with the Engineering team, prior to the formal sign off.
  - **When:** 1 week before the pre-Release sign off
- QA formally signs off the feature pre-Release
  - **When:** by end of week #4 (6-week cycle), by end of week #5 (7-week cycle) or by end of week #6 (8-week cycle) -- or 2 weeks before merge day
  - **References:** <https://goo.gl/AEY99U> (sign off template)

# Time table

## 6-week release cycle

WEEK NO.	WEEK #1 (1st new Beta)	WEEK #2	WEEK #3	WEEK #4	WEEK #5	WEEK #6 (1st RC)
NIGHTLY		Preliminary feature status report	Mid Nightly feature sign offs	Preliminary feature status report	Pre Beta feature sign offs	
BETA			Preliminary feature status report	Pre Release feature sign offs		

## 7-week release cycle

WEEK NO.	WEEK #1 (1st new Beta)	WEEK #2	WEEK #3	WEEK #4	WEEK #5	WEEK #6	WEEK #7 (1st RC)
NIGHTLY		Preliminary feature status report	Mid Nightly feature sign offs		Preliminary feature status report	Pre Beta feature sign offs	
BETA				Preliminary feature status report	Pre Release feature sign offs		

## 8-week release cycle

WEEK NO.	WEEK #1 (1st new Beta)	WEEK #2	WEEK #3	WEEK #4	WEEK #5	WEEK #6	WEEK #7	WEEK #8 (1st RC)
NIGHTLY			Preliminary feature status report	Mid Nightly feature sign offs		Preliminary feature status report	Pre Beta feature sign offs	

BETA					Preliminary feature status report	Pre Release feature sign offs		
------	--	--	--	--	---	----------------------------------	--	--

## Best practices

- **Each feature's QA owner should have a peer assigned to help.** Larger, more complex features can justify more than one QA peer.
- **Internal Test Plan reviews and updates occur periodically.** Feature Test Plans should be updated at least once a week to keep them relevant.
- **Feature status updates should be provided periodically in the QA status documents associated to each Firefox version.**
- **Weekly checks should be made for the bugs reported in the wild.** Since there are so many environment variations (due to various software and hardware pairings), some bugs might only be uncovered by users that have very specific environment setups.
- **A continuous monitoring process should be in place for new bug fixes.** This can be easily done by setting up Bugzilla queries, or something similar.
- **In the case of highly complex features, a meta bug should be created to track all the issue reported by QA.** Having a separate meta bug for the issues reported by QA ensures a more efficient tracking, referencing and reporting.
- **Highly severe bugs (critical, blockers) affecting a feature should be flagged using the qablocker keyword.** Using this keyword in addition to setting needinfo? flags for the right people is the most efficient way of raising major concerns.

## Sign off overview



QA's sign off is based on the quality status of the feature in question. The green, yellow or red quality status is a schedule-based conversation, *not* a general release-readiness status.

## Quality status evaluation criteria

The following aspects are taken into account when evaluating a feature's quality status:

- percentage of completed implementation, based on Engineering's commitment
- unresolved bugs count and severity
- failed tests count and failure reason
- blocked tests percentage and blocking reason

# Quality statuses

## Green

### Status description

A **GREEN** quality status indicates a feature that meets all of QA's expectations.

### Status reasons

- Everything is going according to our plan.
- Manual test results have met the acceptance criteria that was agreed upon with Engineering.
- All or major and critical test objectives agreed upon with Engineering have been covered.
- There are no issues severe enough to jeopardize the release of the feature.

## Yellow

### Status description

A **YELLOW** quality status indicates a feature of questionable quality, meaning that there are issues or potential issues that need to be addressed by the Product, User Experience and Engineering teams as part of that feature's Go/NoGo decision.

### Status reasons

- Manual test results have *mostly* met the acceptance criteria that was agreed upon with Engineering, with a few minor to medium exceptions that need to be addressed.
- Only major and critical test objectives that were agreed upon with Engineering have been covered, minor ones have been left out.
- There are *moderate issues* that might jeopardize the release of the feature, but an input is required from Product, User Experience and/or Engineering to determine if they should block or not.
- Testing progress is behind the planned timeline and we are not able to cover everything on time.

### Example

SUBJECT	[57] [desktop] [feature] e10s with a11y - pre-Release QA sign off
---------	---

## BODY

### e10s with a11y

Targeted GA: Firefox 57 - November 14, 2017.

Sign off phase: pre-Release (Fx57).

Quality status: **YELLOW**

- **Why is this feature yellow?**

- Fx 57 freezes for several minutes at a time when using JAWS v18.0.4321 and v18.0.4350, rendering it unusable and blocking us from testing it properly.
  - This issue was not spotted during the preliminary tests, because we did not have a JAWS license available at that point.
  - Other browsers (Edge, Chrome, Opera) seem to work fine with both of these versions. A screencast showing the behavior on Fx 57 is available here.
  - Our assessment is that while this feature did not improve compatibility with JAWS on Fx 57, it didn't decrease it either, as Fx 56.0.2 shows the exact same behavior.
- [Bug 1411188](#) is also a point of concern for NVDA in particular (see [Comment 1](#) per jteh), although not severe enough to block this feature from shipping with Fx 57.

- **What needs to be done?**

- [Due Nov 03](#) - [\[aklotz\]](#) [\[rkothari\]](#) Take into account the JAWS compatibility issue and [Bug 1411188](#) for the Go/NoGo decision.

Quality trend:

- mid-Nightly: **RED** (signed off on Aug 30, 2017 - [test report](#))
- pre-Beta: **YELLOW** (signed off on Sep 16, 2017 - [test report](#))

Testing summary:

- Test report: <https://testrail.stage.mozaws.net/index.php?/plans/view/6749>
- New bugs: [Bug 1411188](#)
- Known bugs: none
- Firefox 57 feature QA status: <https://goo.gl/i6q3VA>

**What's this report about?** Find out here.

Thanks,  
Alexandru Simonca



	Desktop Release QA
--	--------------------

## Red

### Status description

A **RED** quality status indicates a feature of low quality, meaning that there are severe issues that need to be addressed by the Product, User Experience and Engineering teams as part of that feature's Go/NoGo decision.

### Status reasons

- There's a *big mismatch* between manual test results and the acceptance criteria that was agreed upon with Engineering.
- Part of or all major and critical test objectives that were agreed upon with Engineering have not been covered.
- There are *severe issues* jeopardizing the release of the feature or there's a *big mismatch* between what was planned for implementation and what was actually implemented.
- We are unable to cover important parts of the feature, due to resource constraints or hardware/environment requirements that we cannot meet, which means that those areas may have important issues that we don't know about.

### Example

<b>SUBJECT</b>	[57] [desktop] [feature] Form Autofill MVP with Credit Card support and Sync - pre-Beta QA sign off
<b>BODY</b>	<p>Form Autofill MVP with Credit Card support and Sync  Targeted GA: Firefox 57 - November 14, 2017.  Sign off phase: pre-Beta (Fx57).</p> <p><u>Quality status:</u> <b>RED</b></p> <ul style="list-style-type: none"> <li>• <b>Why is this feature red?</b> <ul style="list-style-type: none"> <li>○ Credit Card information sync has not been implemented yet (see <a href="#">Bug 1395028</a>).</li> <li>○ Form Autofill does not work for some of the top shopping sites (see <a href="#">Bug 1398101</a>), such as Groupon and Newegg.</li> </ul> </li> <li>• <b>What needs to be done?</b> <ul style="list-style-type: none"> <li>○ <u>Due Sep 22</u> - <a href="#">[mnoorenberghe]</a> <a href="#">[vchen]</a> <a href="#">[rkothari]</a> To take into account <a href="#">Bug 1398101</a> for this feature's Go/NoGo decision.</li> <li>○ <u>Due Oct 20</u> - <a href="#">[mnoorenberghe]</a> <a href="#">[vchen]</a> To clarify whether Credit Card information sync is still part of this</li> </ul> </li> </ul>

	<p>feature's scope for Firefox 57.</p> <p><u>Quality trend:</u></p> <ul style="list-style-type: none"> <li>no previous sign offs available</li> </ul> <p><u>Testing summary:</u></p> <ul style="list-style-type: none"> <li>Test report: <a href="https://testrail.stage.mozaws.net/index.php?/plans/view/5872">https://testrail.stage.mozaws.net/index.php?/plans/view/5872</a></li> <li>New bugs: <a href="#">Bug 1399065</a>, <a href="#">Bug 1399402</a></li> <li>Known bugs: <a href="https://mzl.la/2CPaUBI">https://mzl.la/2CPaUBI</a></li> <li>Firefox 57 feature QA status: <a href="https://goo.gl/i6q3VA">https://goo.gl/i6q3VA</a></li> </ul> <p><b>What's this report about?</b> Find out here.</p> <p>Thank you, Gabi Cheta Desktop Release QA</p>
--	--

## Pref'd OFF

### Description

Features riding a specific Firefox version in a disabled state (via preference flip) will not be receiving a sign off report from QA. This is because there's no real impact for the end user, and so an in-depth assessment of that feature's quality is irrelevant for the Firefox version in question.

Instead of a sign off, QA will send out a regular test report describing what was considered in scope/covered.

### Example

<b>SUBJECT</b>	[desktop] [mobile] [feature] Structured Sync Bookmark Application - Test Results
<b>BODY</b>	<p>Hello,</p> <p>I have completed the testing for Structured Bookmarks Application feature. Please see below the details:</p> <p>The testing was done on following platforms:</p> <ul style="list-style-type: none"> <li>Mac OS 10.13</li> </ul>

- Windows 10
- Ubuntu 16.0.4
- iOS
- Android

Testing summary:

- Test Plan: [https://docs.google.com/document/d/15JYYcDCz\\_wDohD-6gcwnNhqn\\_seWfHP\\_CfXfdKcH2eA/edit?usp=sharing](https://docs.google.com/document/d/15JYYcDCz_wDohD-6gcwnNhqn_seWfHP_CfXfdKcH2eA/edit?usp=sharing)
- Test report: <https://testrail.stage.mozaws.net/index.php?/reports/view/824>
- New bugs: <https://mzl.la/2ytVEIV>

**What's this report about?** Find out [here](#).

Please note that these are just the testing results for the verification that we completed recently on Nightly 60. It is not a formal sign-off as the feature will be pref'd off by default in beta 60. Please feel free to contact me in case of any questions/clarifications.

Thank you,  
Kanchan Kumari  
Engineering QA

## Sign off template

The feature sign off template is available [here](#).

## Useful references

- All QA documentation available for features, structured per Firefox version: [here](#).
- Feature QA status per Firefox version:
  - Firefox 66: [here](#)
  - Firefox 65: [here](#)
  - Firefox 64: [here](#)
  - Firefox 63: [here](#)
  - Firefox 62: [here](#)

- Firefox 61: [here](#)
- Firefox 60: [here](#)
- Firefox 59: [here](#)
- Firefox 58: [here](#)
- Firefox 57: [here](#)