



Submit metrics

Tags: [Flow](#)

The submitter metrics make it easy to understand how PR submitters are responding to and incorporating feedback from the code review process.

In this article

[Submit Metrics](#)

[Responsiveness](#)

[Unreviewed PRs](#)

[Comments Addressed](#)

[Receptiveness](#)

Who can use this?

Core

Plus



Submit Metrics

The Submit metrics include:

- **Responsiveness:** How long it takes a submitter to comment or revise code in response to a comment on their PR.
- **Unreviewed PRs:** How often submitted pull requests get no comments.
- **Comments Addressed:** How often a submitter responds to a reviewer's comment.
- **Receptiveness:** How often a submitter accepts reviewer feedback and revises their code.

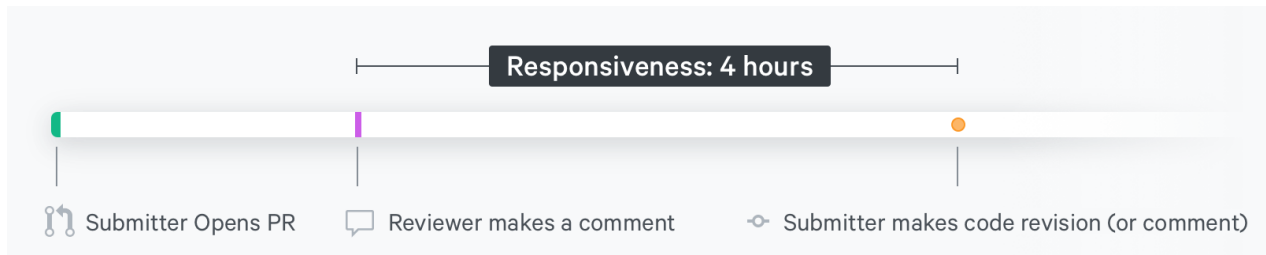
Note: Comments addressed and Receptiveness metrics are only available in Flow Enterprise Server.

These metrics are designed to promote healthy collaboration and provide prescriptive guidance to improve the productivity of the team's code review process as a whole. As with any data point, these metrics should be used in context. What's right and what's normal depends on your team's situation.

[back to top](#)

Responsiveness

Responsiveness lets you know if people are responding to feedback in a timely manner.



Responsiveness is the average time it takes to respond to a reviewer's comment with either another comment or a code revision. It looks at the time between the last comment of the reviewer and the submitter's response.

In practice, the goal is to drive this metric down. It's up to the manager to determine how quickly people should respond to comments, but depending on your deployment frequency or deadlines, you may find that less than four hours is ideal, while over 24 hours regardless of timezone is counterproductive under most circumstances.

However, like everything we do, Responsiveness is context-dependent.

The submitter may be in the zone and shouldn't stop. In some cases, it may be inappropriate for them to stop, such as attending meetings, working on an extremely important ticket, or handling an outage.

As soon as you exit your flow state, like breaking for lunch or coffee, you should take the time to try to respond to those comments.

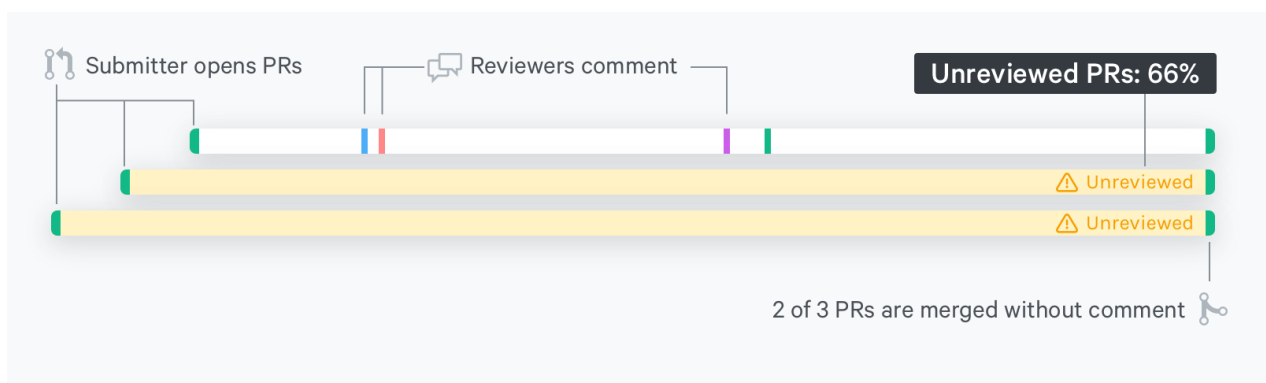
A response is either a comment or a code revision. Both options are viable responses.

Learn more about [Responsiveness](#).

[back to top](#)

Unreviewed PRs

Unreviewed PRs lets you know if PRs are getting the proper level of review.



The **Unreviewed PRs** metric shows the percentage of PRs that didn't get any reviews.

In an ideal world, PRs are never merged without being reviewed. Even when they're small or made by senior developers they can be a huge source of bugs. Many organizations establish a policy and configure their system to programmatically reject unreviewed PRs.

But for those who don't enforce a policy, every single time the PR goes out without being reviewed, a manager should know.

As a manager, you should drive this number to zero and take seriously the rare instance when an engineer felt compelled to drive code straight from the laptop to production without anyone ever looking at the change.

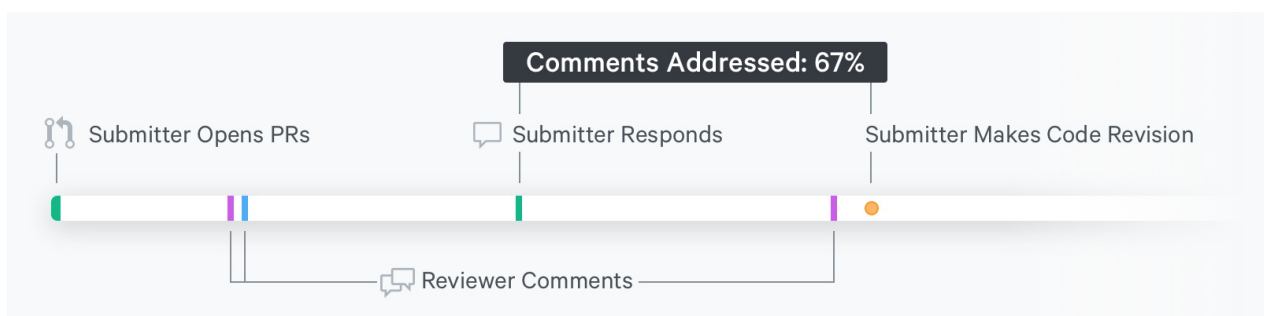
Learn more about [Unreviewed PRs](#).

[back to top](#)

Comments Addressed

Note: Comments addressed is only available in Flow Enterprise Server.

Are people acknowledging feedback from their teammates?



Comments Addressed is the percentage of reviewer comments the PR submitter responded to with a comment or a code revision.

This metric is different from Responsiveness because it looks at how broadly the submitter responded to the reviewer's comments instead of how quickly they responded to them.

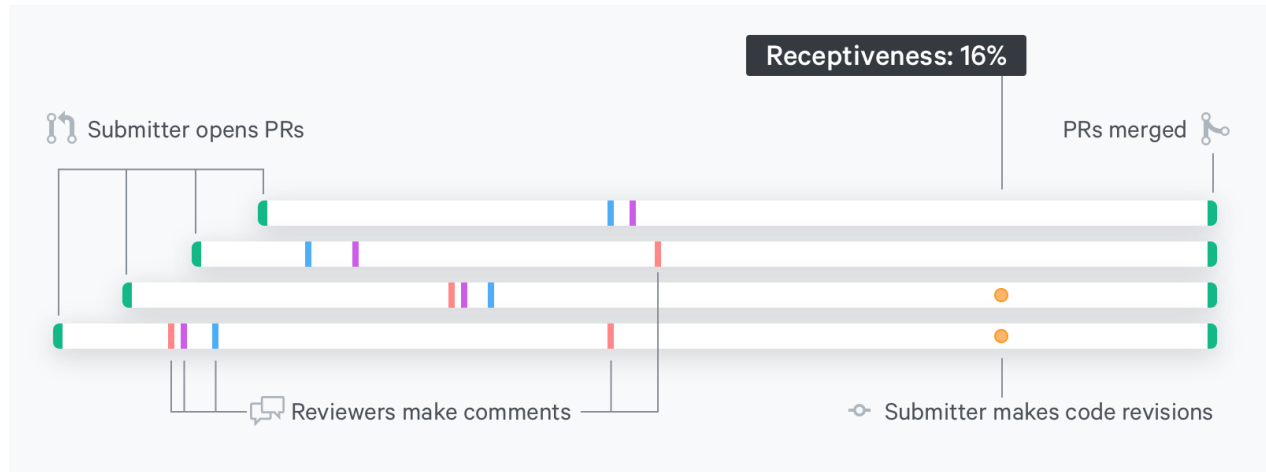
As a manager, you want to drive this number up. If a reviewer thought it was worthwhile to make a comment, it's generally worthwhile to respond to it. It's best to use this metric as a prompt to encourage thorough reviews rather than managing to an absolute target.

[back to top](#)

Receptiveness

Note: Receptiveness is only available in Flow Enterprise Server.

Receptiveness shows you if people are incorporating feedback from their teammates.



Receptiveness is the ratio of follow-on commits to comments. In short, this metric looks at whether the PR submitter is taking people's feedback and incorporating that into their code.

This is a [Goldilocks metric \(external site, opens in new tab\)](#), so you'll want to manage the outliers — and as always, context matters. A good developer is always open to improvements, but not all suggestions are worth implementing.

If Receptiveness is too low, it could be a sign that a developer is closed to any input regardless of merit. It could also be a sign of "rubber-stamping," which is a work pattern where the reviewer assumes too much of the submitter and approves the PR without a thorough review.

If Receptiveness is too high, you may be seeing a developer failing to stand their ground, or someone relying on the review process to shake out bugs easily caught in development.

If you need help, please email support@pluralsight.com for 24/7 assistance.