

Review Metrics

Tags: [Flow](https://pluralsight.knowledgeowl.com/help/search?phrase=:Flow) (<https://pluralsight.knowledgeowl.com/help/search?phrase=:Flow>)

Review metrics provides a way for software teams to see the ground truth of what's happening in the code review process. The package is split into three sets of metrics: [Submit](https://help.pluralsight.com/help/submit-metrics) (<https://help.pluralsight.com/help/submit-metrics>), [Review](https://help.pluralsight.com/help/team-collaboration-metrics), and [Team collaboration](https://help.pluralsight.com/help/team-collaboration-metrics) (<https://help.pluralsight.com/help/team-collaboration-metrics>).

Who can use this?

<u>Core</u>	<u>Plus</u>
✓	✓

The four Review metrics, found in the **Review collaboration** report, include:

- **Reaction time:** The time it takes for the reviewer to respond to a comment addressed to them.
- **Involvement:** The percentage of pull requests that the reviewer participated in.
- **Influence:** The ratio of follow-on commits made after the reviewer commented.
- **Review coverage:** The percentage of hunks commented on by the reviewer.

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" will vary depending on your team's culture.

In this article

[Reaction time \(\)](#)

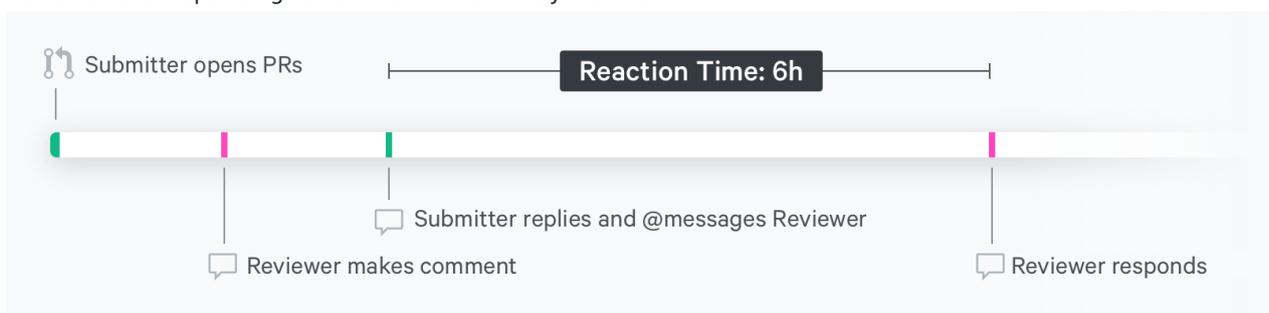
[Involvement](#)

[Influence](#)

[Review coverage](#)

Reaction time

Are reviewers responding to comments in a timely manner?



Reaction Time is the time it takes for a reviewer to respond to a comment addressed to them. Reaction Time for the reviewer is the same concept as [Responsiveness](https://help.pluralsight.com/help/submit-metrics) (<https://help.pluralsight.com/help/submit-metrics>) for the submitter.

In practice, the goal is to drive this metric down. You generally want people to be responding to each other in a timely manner, working together to find the right solution, and getting it to production. If someone addresses you directly, you want to respond to them within an hour or so, and more than eight hours is usually counterproductive under most circumstances.

However, like everything we do, Reaction Time is context-dependent.

An engineer may be in the zone and shouldn't realistically stop. In some cases, it may be inappropriate for them to stop (they're in a meeting, working on an extremely important ticket, or handling an outage).

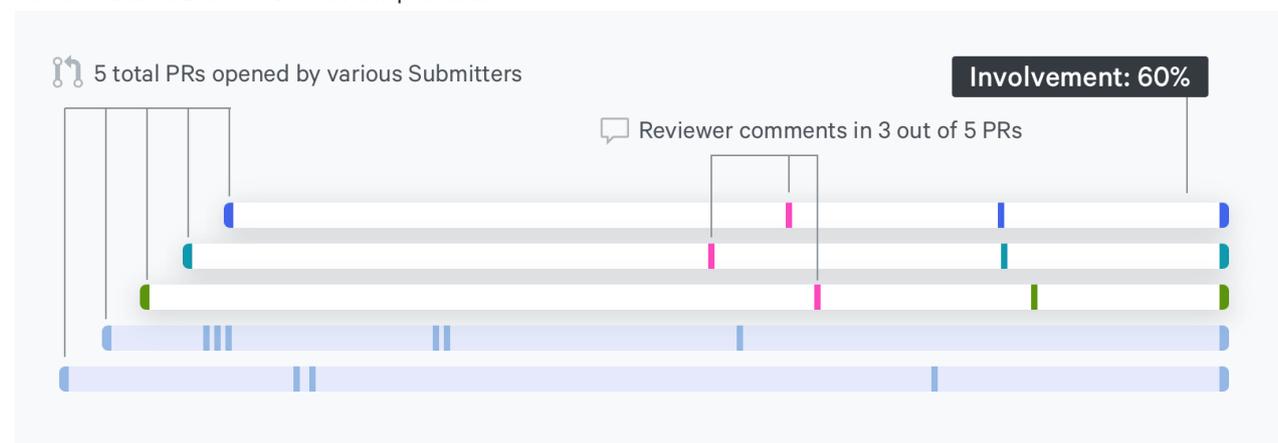
But when it's a your work versus their work situation you should take the time to respond to those comments as soon as you exit your flow state, like when breaking for lunch or coffee.

[back to top](#)

Involvement

Note: Involvement is only available in Flow Enterprise Server.

Involvement shows how involved each person is in reviews and gives insight into whether some people are more involved than others in the review process.



This number will change according to your view. If you're looking at their home team, an individual may show 75% involvement, indicating they reviewed three out of four PRs. But if you zoom out to view the whole organization, that same individual's involvement rate will be much lower.

Involvement is very context dependent. Not everyone can review everyone else's code. For example, you wouldn't make an HTMLer review a complex query optimization. Architects and team leads are usually expected to have more Involvement to ensure consistency.

However, you should find a [Goldilocks zone](https://en.wikipedia.org/wiki/Goldilocks_principle) (https://en.wikipedia.org/wiki/Goldilocks_principle) for each individual and the team they're on and manage significant or sustained changes to their Involvement.

The **Review collaboration** report shows Involvement for groups of individuals, organizations and teams. The calculations for these are worth a special mention.

For a team, Involvement helps you understand how often the team reviews their teammates PRs as opposed to outsourcing to another team. Having extra people, like architects, review a PR is great, but you don't want it to replace team reviews.

For example, this metric is helpful if you have teams set up by role, for example separate teams for Software, DBA, QA, front end, and Back end, but your project teams are cross-functional. You likely want someone from the organizational team to review a PR for standards and best practices and someone from the project team to make sure the work meets business objectives.

Setting up an organizational team structure and a project team structure and making sure that both have a high Involvement is a good way to accomplish this.

The calculation for team Involvement is calculated like this:

$$[\# \text{ PR's reviewed by a team member}]/[\# \text{ reviewed PRs}].$$

For your entire organization, Involvement measures how often someone in the organization reviewed the PRs. Once your teams are set up properly and everyone in your organization is on the team, Involvement becomes:

$$[\# \text{ reviewed prs}]/[\# \text{ total PRs}]$$

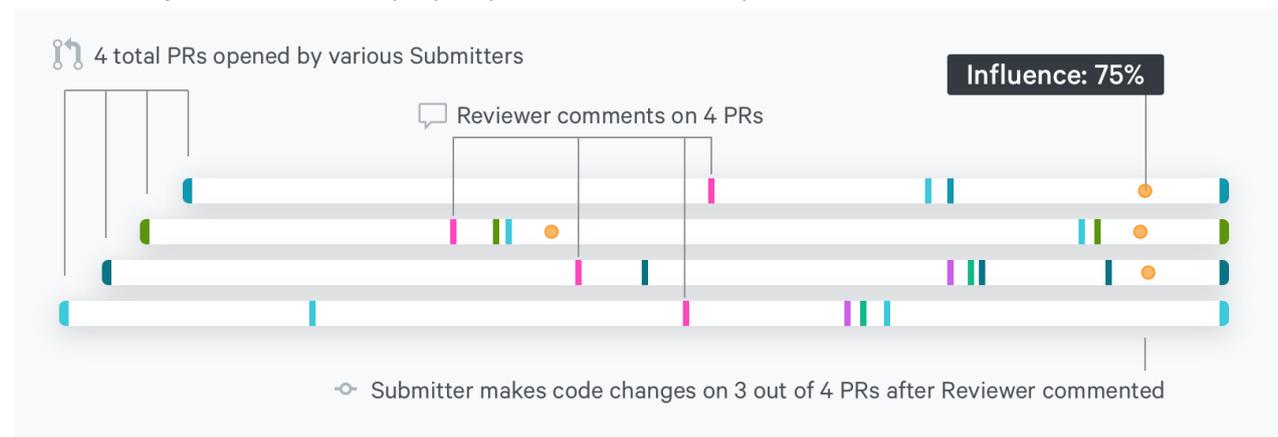
This equation is the inverse of Unreviewed PRs.

[back to top](#)

Influence

Note: Influence is only available in Flow Enterprise Server.

Influence lets you know how often people update their code in response to reviewer comments.



Influence is the ratio of follow-on commits made after a reviewer posted a comment. It's the sibling of

Receptiveness (<https://help.pluralsight.com/help/submit-metrics>). The Influence metric looks at whether your comments elicited a follow-on commit.

Influence doesn't try to assign specific credit. That is to say, no one person gets the credit for being influential. We understand that it's the discussion itself that deserves the credit, so all participants in the discussion prior to the follow-on commit get Influence credit counted toward the metric.

In practice, there's a middle ground with this metric: Low influence may be a signal that an individual isn't making substantive comments. High Influence may be a signal that an individual is acting as a gatekeeper or a crutch.

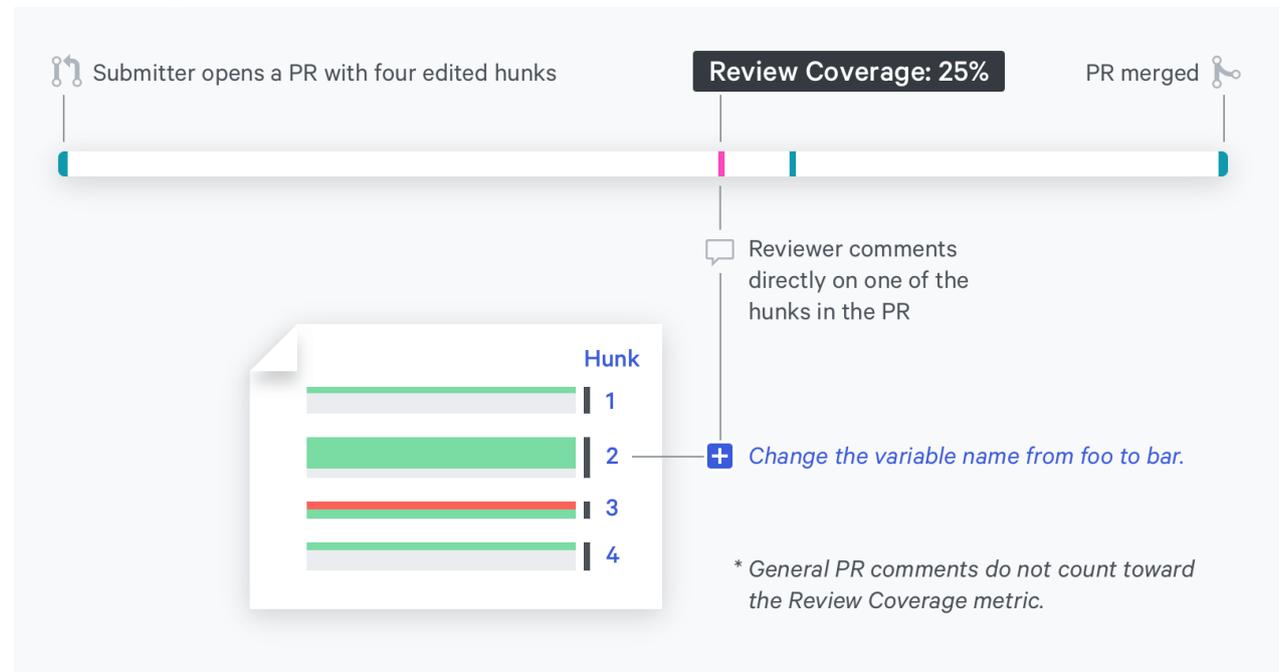
Architects and team leads should have higher Influence metrics. Once you find the right level for each individual and team, manage significant or sustained changes as they could indicate a shift in the team dynamic that warrants a manager's attention.

[back to top](#)

Review coverage

Note: Review coverage is only available in Flow Enterprise Server.

Review coverage shows how much of each PR has been reviewed.



Review Coverage is the number of hunks in a PR commented on as a percentage of the total hunks in the pull request. A typical PR will contain multiple files and multiple edits (aka hunks) on that file.

Like a teacher who puts tic marks on every page of a term paper to indicate they read it, a good reviewer will put a comment on the majority of the edits of a PR even if it's a simple "LGTM". In practice, 100% Review Coverage is overkill.

As a manager, you want to keep an eye on Review coverage. When coverage rises and falls at the individual or team level, you'll know to provide guidance. Encouraging team members to take the time to review each change in the code and not just skim it as a whole will drive this number up. Small changes in average Review Coverage make a big difference.

[back to top](#)

If you need help, please email [Support \(opens email form\) \(\)](#) for 24/7 assistance.