Snapshot

Who can use this?

Core   Plus
        ✓

> **Note:** The Snapshot report is only available to Flow Enterprise Server users. All others should use our new Check-in report ().

The **Snapshot** report provides a summary of an individual developer's work patterns for a period of time. The report also gives one specific piece of feedback for the developer, which can be useful for one-on-ones or reviews.

What's important in this report is not an engineer's position at any specific moment, but how their position changes over time. (We recommend 2 week time-spans). Developers who make contributions one week might be contributing in other ways the following week.

In this article

[What do the different quadrants mean in the Snapshot report?](#)

[How does the feedback work in the Snapshot report?](#)

What do the different quadrants mean in the Snapshot report?

The chart intelligently plots an engineer's contributions across 2 axes:

- Impact (https://pluralsight.knowledgeowl.com/help/what-is-impact) is the severity of edits to the codebase, as compared to repository history.

- Rework (https://help.pluralsight.com/help/what-is-churn) is code which was thrown away shortly after being written.
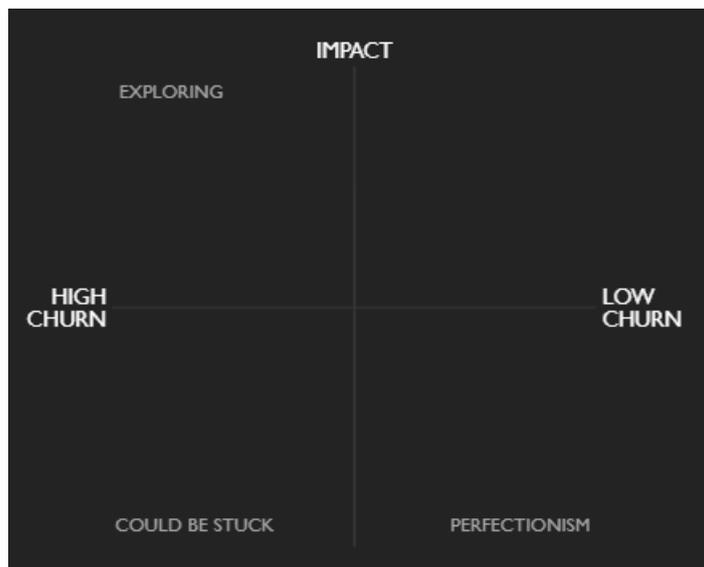
The bottom two quadrants usually are typical for part-time engineers, team leads, managers, QAs, etc.**If you have full-time developer fall in the bottom two quadrants—especially the bottom-left 'could be stuck' quadrant—they may need some help.**

Exploring (a.k.a. discovery)

Developers in this quadrant might be working with a new language or experimenting with a new type of feature, or trying out many implementation paths. Occasionally, engineers in this quadrant haven't been provided with clear specs or requirements, which is why they have high rework.

Prolific

Developers in this quadrant are making exceptional contributions. So long as the developer has the correct requirements (i.e. building the right thing), interrupting them might be costly. They are building impactful code that doesn't rework.

Could be stuck

If an engineer is in this quadrant they might need some help on a feature. It might be worth checking in, especially if you see them in this quadrant week over week. You don't want full time engineers to be here for very long, so you can use this report to check in and help before it's too late. Sometimes Q/A and Product Managers appear in this quadrant. If you'd like to exclude these irregular Users here's how to hide someone (https://pluralsight.knowledgeowl.com/help/excluding-users-from-reports) from appearing in the reports.

Perfectionism

This is a quadrant you might expect managers, team leads, and other senior roles who are familiar with the codebase. They tend to have lower throughput because they are helping others or leading a team. A full time engineer wouldn't typically want to hang out in this quadrant since they are not making a huge impact to the codebase. However most of the work checked in is sticking around.

For more info on how the developer feedback works, check outhow does the feedback work in the Snapshot report (https://pluralsight.knowledgeowl.com/help/how-does-the-feedback-work-in-the-snapshot-report).

back to top

How does the feedback work in the Snapshot report?

The **Snapshot** report provides a summary of an individual developer's work patterns for a period of time. For more about how the quadrants work, review what the different quadrants mean in the Snapshot report (https://help.pluralsight.com/help/what-do-the-different-quadrants-mean-in-the-snapshot-report).

The report also gives one specific piece of feedback for the developer, which can be useful for one-on-ones or reviews.

There are 10 possible pieces of feedback. The ranking functions sort into a waterfall, with these possible outcomes:

1. Check in your latest progress every day

2. Spend a little more time coding each day

3. Take on some more important tasks.

4. Break large work items into pieces

5. Develop a clear implementation path before starting work

6. Balance large refactors with work on new features

7. Take on some more difficult work

8. Help the team pay down technical debt

9. Review code only after making progress on your own items

10. Honestly, we couldn't find any suggestions. Keep kicking butt.

Here are some examples of how that feedback might be presented:

- An engineer might rework at an unusually high rate relative to the team, so #5 may be relevant (Develop a clear implementation path before starting work). However, if they are also checking in abnormally large commits, they would receive the higher-priority #4 Break large work items into pieces.

- If an engineer's work signature contains mostly low-impact commits, they may receive feedback #7, encouraging them to dig in and take ownership of larger features.

- If an engineer falls substantially lower than the team average for meaningful active days, they may receive #3 Take on some more important tasks.

Why do I see a lot of "Check in your latest progress every day"? Some of what is at play is a Flow bias—that progress should be committed and pushed on a daily basis. This behavior is something we've found to be particularly good for productivity (https://www.pluralsight.com/blog/teams/prolific-engineers-take-small-bites--patterns-in-developer-impac) across the 20+ million commits we've analyzed.

As a result, anyone who is under 3 days per week will hit this feedback in the waterfall logic, regardless of the team average.

back to top

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