



Dev weeks best practices

Tags: **Flow**

Use this article for additional information on how to best use [dev weeks](#) in your organization.

In this article

[Best practices for understanding dev weeks](#)

[Dev weeks and time](#)

[Dev weeks accuracy](#)

Who can use this?

Core

Plus



Best practices for understanding dev weeks

Developing the algorithm

- When developing this algorithm, Flow teams manually calculated the percent of effort spent on tickets at the end of every week. We also ran the dev week algorithm on the same data to see how well it represented the truth of how developers understood their work.
- Through weekly and rigorous testing, we implemented the activity signals in the algorithm and adjusted them until we calculated a depiction of effort that resonated with our developers.
- We also worked with several customers who already tracked their effort to understand how our algorithm matched their current truth. Some organizations were able to stop their previous time tracking methods after using our algorithm.

Note: Flow doesn't account for non-development-related activities like meetings and documentation in the dev weeks algorithm.

Using dev weeks with other Flow views

Dev weeks views can be similar to Story points and Ticket views in [Investment profile](#), but there are some important differences to understand:

- Unlike Tickets and Story points views, dev weeks accounts for all ticket states, while Tickets and Story points only look at Done work.

- Tickets and Story points aren't measurements of effort, and shouldn't be used to measure effort.
 - For example, consider a developer who works on two tickets during a week. The Tickets view assumes a 50/50 distribution of effort on each ticket. With dev weeks, Flow can identify if the effort on one ticket amounts to 80% of their activity for the week. This tells a very different story about how developers are spending their time.
- Use the Ticket and Story point views in conjunction with dev weeks to tell a fuller picture of how your developers are using their resources.

[back to top](#)

Dev weeks and time

One of the biggest questions we get about dev weeks is whether it tracks a developer's time. The short answer is: it doesn't. Dev weeks is a measure of effort, not time.

Dev weeks doesn't track the exact number of hours spent on work. There's a lot of work that goes into software development that can't be traced exclusively from ticket and git signals.

Instead, we use a probabilistic algorithm to combine the activity signals ingested from ticket and git sources to help us understand where and when activity is happening. The algorithm proportionalizes the amount of effort spent within a week to help you understand where effort is happening for individuals, and then displays that information at a team and organization level.

As an example, let's consider a developer who only works one day during a week. We look at all activity in that week, then proportionalize it. Our algorithm says that the developer's weekly allocation was spent on whatever the developer worked on, whether it's one ticket or ten.

Tip: Sometimes people wonder if a dev week is the same as 40 hours. That may be true, but isn't always. A dev week represents the work a developer did during a given time period. It could be 20 hours of work or 60 hours of work, depending on the organization. We recommend that you have a conversation as an organization about what a dev week means to you if you're going to use the information for feature costing or making plans for the future.

[back to top](#)

Dev weeks accuracy

If you don't already use some sort of timecarding system to track dev weeks, it may be hard to quickly check the accuracy in the data. At this time, the best way to do this is to show the Flow data to your teams and ask them to provide feedback on what the dev week algorithm produces. Once your teams approve of the numbers, we hope you'll be able to trust it moving forward.

Tip: In having these conversations with our own developers, we noticed they often undercounted or

forgot the effort they spent on commenting on others' PRs and tickets. Dev weeks includes activity signals from comments on others' work, so they get credit for that work. If you're noticing discrepancies from what you or your developers expect, check to see if you're including their commented work in your expectations.

To help increase accuracy, one thing you can do is make sure all your unassigned work in Flow is assigned to a Work type and Investment layer. Having work assigned will make your dev weeks data much more useful. Not sure where to start? Begin by tagging your epics in Jira with the correct Investment layer so all tickets associated with the epic inherit the Investment layer unless otherwise specified. Learn more about [configuring Work types and Investment layers](#).

Note: If you or your teams see something that looks inaccurate, reach out to Support so we can understand what types of workflows your team uses and why your data may appear incorrect.

Tips for improving dev week accuracy

While the algorithm can give accurate results without you needing to set up a lot of configurations, there are some steps you can take to increase its accuracy.

- Without ticket data from Jira, Flow cannot display any dev week data.
- Flow works to link your PRs and commits with your Jira issue. The more directly these are linked in your system, whether through putting ticket numbers in commit messages or adding a link to the commit in the Jira issue, the more reliably Flow can extract data from the signals to give you accurate dev week calculations.
- Make sure your integrations are processing correctly without interruption. If Flow loses the connection with your integrations, dev week calculations may be inaccurate. As more data is processed, your dev week numbers will change as new signals are detected.
- Double-check all your [ticket project configurations](#) to make sure all ticket types that should count toward dev activity are included, not excluded. Activity, including commit and PR activity, associated with excluded tickets don't factor into the dev week calculation.

Note: If you change a previously-excluded ticket type to included, all historical metrics will change to include that ticket type. This affects all ticket metrics, not just dev weeks.

- When checking your ticket project configurations, make sure your waiting states are configured. Having accurate waiting states is essential to Flow correctly calculating Active time, which is one of the most important signals for dev weeks.
- Configure Investment layers and Work types in your ticket projects. You'll see dev week data for unassigned tickets, but without configuring Investment layers and Work types, you won't be able to see proportions of effort across these categories. This reduces your ability to understand how your dev weeks are spent across initiatives.

- In your systems, make sure you're updating tickets as changes are happening. Update your statuses, assignees, and more to make sure Flow quickly gets the most accurate representation of the work you're doing. If your tickets aren't up-to-date, your dev week calculations may not represent actual work being completed.
- Flow includes in-progress tickets in calculations by applying the end date of the selected period as the theoretical end date of the ticket. As tickets move to their final Done status, dev weeks may change.
- Make sure your tickets are associated with epics for an epic-level view of how many dev weeks it takes to complete an epic.
- If a user is on multiple teams in Flow, their dev weeks will count the same for all teams they're part of, regardless what work they're doing.

If you need help, please contact [Pluralsight Support](#).