Use Check-in to get concise, data-driven status updates on engineers’ work before and during 1:1s.

Check-in has two views. Set goals for your users with Status update and get quick insights into an engineer’s progress. Player card tracks a user's trends across Flow metrics.

Who can use this?

<table>
<thead>
<tr>
<th>Core</th>
<th>Plus</th>
</tr>
</thead>
</table>

Use both to help 1:1s run more efficiently and focus the conversation on things that matter to you and your team.

In this article

- Selecting a user and date range
- Status update overview
- Setting a user’s short term goals
- What type of commits has a user focused on lately?
- How complex are a user’s commits?
- How does a user balance coding and reviewing?
- How does a user balance impact and efficiency?
- Viewing a user’s total activity
- Viewing a user’s PR activity
- What does Player card show?
- How do I customize Player card’s metrics?

## Selecting a user and date range

Select a user and date range for insight on where that user is doing well and where they can improve. These settings apply to the data of Status update and Player card.

To select a user, date range, and team:

1. Click the down arrow next to the current user’s name to select your desired user. Search by their name, or filter for a user by recent users, users on your teams, or all users.
2. Click the time period next to Date range. Select a preset date range or use the calendar to create your own date range.

3. Click on the team filter to select the team to compare this user to. Choose whether nested teams are included or excluded.

**Note:** For PR-based metrics, selecting a different team may change the user's metrics. PRs are only included in metrics if they were created by a user on the currently-selected team.

---

**Status update overview**

Use Status update to help set short term goals for a developer, celebrate their successes, and suggest areas of improvement. Status update provides quick insights into a developer's progress on their goal by measuring their activity within a selected time period. Status update breaks down a user's work patterns across commits, code review, ticket activity, and PRs.
Setting a user’s short term goals

In Status update, users can create short term goals. These goals help define a user’s success and guide week-over-week improvement on metrics that matter to the user.

**Note:** Any user with access to Check-in can edit their own personal short term goals. We suggest that only managers and team leads have access to edit others’ goals.

Each objective corresponds with a specific metric:

<table>
<thead>
<tr>
<th>Goal</th>
<th>Metric(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streamline workflow and remove blockers like unbatched meetings to</td>
<td>Coding days</td>
</tr>
<tr>
<td>have more time to create solutions in the code base.</td>
<td></td>
</tr>
<tr>
<td>Make small, more frequent commits to make commits easier to</td>
<td>Commits per day</td>
</tr>
<tr>
<td>review and enable CI/CD.</td>
<td></td>
</tr>
<tr>
<td>Clarify requirements and scope up front to increase the quality</td>
<td>Efficiency</td>
</tr>
<tr>
<td>and maintainability of written code.</td>
<td></td>
</tr>
<tr>
<td>Reduce risk by reducing the number of PRs that are merged without</td>
<td>Unreviewed, PRs</td>
</tr>
<tr>
<td>being reviewed.</td>
<td></td>
</tr>
<tr>
<td>Maintain focus on new features by spending more time committing</td>
<td>New work, Legacy</td>
</tr>
<tr>
<td>new code while maintaining a small and steady commitment to</td>
<td>refactoring</td>
</tr>
<tr>
<td>rewriting/cleaning old code.</td>
<td></td>
</tr>
<tr>
<td>Focus on rewriting old code and helping coworkers with their code</td>
<td>Legacy refactoring,</td>
</tr>
<tr>
<td>to improve the overall quality of the codebase.</td>
<td>Helping others</td>
</tr>
<tr>
<td>Remove bottlenecks (like long running asynchronous communication</td>
<td>Time to merge</td>
</tr>
<tr>
<td>between devs) in the PR review process to enable timely merges of</td>
<td></td>
</tr>
<tr>
<td>PRs into the codebase.</td>
<td></td>
</tr>
</tbody>
</table>

To set short term goals:

1. Click **Edit goals** in the top right corner of short term goals.
2. Click the Objectives dropdown arrow. Select a goal from the Objectives dropdown.

3. Add a target value to help track progress of a user’s objective over time.

**Tip:** Enter targets that are achievable in the time between 1:1s. Goals should be small enough that you see incremental improvement between 1:1s, but large enough that you challenge your team members to grow over time.

**Note:** If the user performs at or above the team average in the selected objective, the Check-in report encourages the user to select a different objective.

4. Click **Save goal**. The goal and current progress appear in the Status update tab at the bottom of short term goals.
When a user hits their goals, edit their goals to challenge them with new goals. Click the **Edit goals** button and select a new objective or adjust their current objective’s values. This strengthens team members across goal areas.

What type of commits has a user focused on lately?

In the Status update tab, Commit focus shows the percentage of each work type contributed by a team member in the selected date range. Commit focus categorizes a user’s work according to the following four types: new work, legacy refactor, help others, and rework.

Use Commit focus to balance the type of work a user does on each project. For example, you might expect legacy refactor to be at 55%. If you see that it is below 55%, use your 1:1 to explore why and what you both can do to better set and meet goals.
How complex are a user’s commits?

Commit complexity is a measure of how likely it is that a particular commit will cause problems. This Status update report ingests all of a user’s commits and classifies each as Low, Medium, or High complexity.

Some complexity is expected. Watch for spikes in complexity toward the end of a sprint. Spikes may indicate setbacks to accomplishing your and your team’s goals.

Click Show details to view an in-depth look at the complexity and focus of specific commits.

Filter commits by their complexity. Filter a user’s commits to get a closer look and identify patterns that may warrant follow-up conversations in your 1:1s. Click on a work type column, such as Legacy refactor, to sort the results by work type.
How does a user balance coding and reviewing?

The code/review balance matrix compares a user’s coding days to their number of reviewed PRs. Use this matrix in Status update to ensure your engineers practice a healthy balance between individual contributions and PR review and collaboration.

Toggle between the code/review balance matrix and the codebase contribution matrix by clicking the down arrow next to the metric name.

Hover over the data points to see an exact breakdown of a team member's code/review balance as well as the average code/review balance of whichever team occupies the team filter.

How does a user balancer impact and efficiency?

The codebase contribution matrix compares a user's impact to efficiency percentage. Use this matrix to show the
cognitive difficulty of a set of work and how much rework was required to accomplish it.

Toggle between the code/review balance matrix and the codebase contribution matrix by clicking the down arrow next to the metric name.

Hover over the data points to see an exact breakdown of a team member's codebase contribution as well as the average codebase contribution of whichever team occupies the team filter.

If impact increases, look for efficiency to hold relatively steady. High impact commits should only happen at certain times in the project's lifecycle. Try to avoid high impact commits at the end of a sprint or during the fit and finish stage of a project.

Viewing a user's total activity

Status update includes the Work log view of an individual user to give an overview of their recent work. This helps spot patterns and trends within a user's activity.

Both manager and engineer can review this section before 1:1s. This helps make 1:1s more efficient by keeping you on the same page regarding work status. Learn more about Work log.

Use the Show filter to select which commit types, PR activity, PR comments, or Ticket activity you'd like to see.
Use the Weekends toggle to choose whether the chart displays activity from weekends. Look at activity from other weeks by clicking the previous page and next page arrow buttons on the left and right. Hover over an icon to view more details about that activity.

**Viewing a user’s PR activity**

The PR section shows the PRs open during the selected time frame. Managers can use the PR section to gain insight into open, long-running PRs and see where an engineer needs assistance. Once the manager understands how to help their team, they can provide help and coaching.

Narrow down which PRs you see by choosing a Sort by option, like Recently opened, Biggest, and Least activity. Using these filters means you see exactly which pull requests you're interested in.

Hover over a pull request to view more information about it, such as recently committed code or comments. Click on a PR to see the pull request's number and a link to view the pull request itself.

**What does Player card show?**

Player card displays metrics across four categories relevant to assessing a developer's overall contributions. The four metric categories are culture, activity, predictability, and efficiency. Each category has a colored border that correlates with the category legend at the top of Player card.

---

**Note:** View rights impact what data you see in Player card. For collaborative metrics like Iterated PRs, Reaction time, PR iteration time, Thoroughly reviewed PRs, Time to first comment, and Time to merge, your Player card data will look different if you only have access to view your own metrics rather than having access to view other's metrics. Having limited view rights limits which actions can be counted toward your collaborative metrics in Flow reports.
Culture

Culture metrics pertain to a developer’s role within their larger team. This gives insight on how the developer contributes to the code review process.

Activity

Activity metrics reflect how a developer’s habits contribute to their team’s work pace.

Predictability

Predictability metrics measure how efficiently your team completes their work.

Efficiency

Efficiency metrics measure how a developer contributes to the overall flow of the team and the rate they move work through the pipeline.

Additional insight on individual metrics

To get more information on an individual metric, click Show details within a metric card.

Expanding a metric shows a developer’s target for that particular metric and the developer’s current status on that target. Learn more about setting targets.
How do I customize Player card’s metrics?

You can customize Player card to display the metrics that most matter to you in your 1:1s.

To customize which metrics Player card displays:

1. Click **Edit metrics** in the top right of the Player card report.
2. Select the metrics you want to include in your Player card report.
3. Click **Update layout**. Your selected metrics appear within the Player card report.

If you need help, please contact [Pluralsight Support](mailto:pluralsight-support@pluralsight.com).