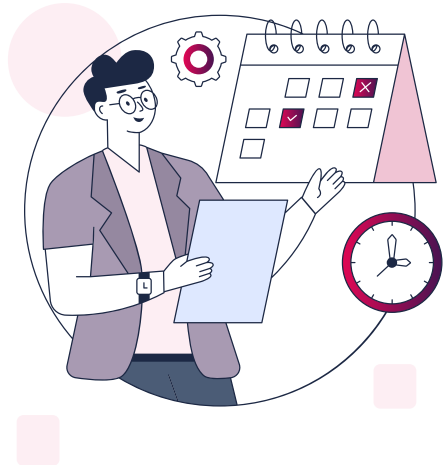


Time in Status Reports for Engineering Managers



If you manage a dev team on Jira, you've had this conversation before: sprint velocity looks fine on paper, but delivery still feels slow. Tickets close, standups sound normal, and yet the release keeps slipping. Everyone's "working," but nobody can say exactly where the time actually went.

The problem isn't your team. It's that Jira's default view only shows you whether work moved, not how long it sat at each stage before it did.

Time in Status Reports by RVS Softek gives engineering managers what story points and burndown charts can't, a status-by-status breakdown of how every issue actually moves through your workflow, so you can see cycle time by assignee, spot rework before it snowballs, and back up delivery conversations with data instead of gut feel.

The Real Cost of Invisible Cycle Time

Most engineering teams measure output at the sprint level: points committed vs. points closed. Very few measure where an issue's time is actually spent between "To Do" and "Done."

A ticket picked up on Monday and closed on Friday looks like a five-day cycle. But if it sat untouched in "In Progress" for two days before a developer picked it up, then bounced back from "In Review" twice because of unclear requirements, the real story is a lot messier than the burndown chart shows.

Without status-level visibility, engineering managers keep having the same reactive conversation: a release slips, and only then does anyone dig into why. With it, you can see the pattern forming in real time, which stages are consistently slow, which issues keep bouncing back, and where to step in before it affects the sprint.

What do Engineering Managers Need?

Jira tracks status changes.

But out of the box, it doesn't tell you:

- ✦ How long each issue actually sits with a given assignee in each status
- ✦ Whether cycle times are getting better or worse over time, not just this sprint, but the last few months
- ✦ Which issues are cycling back through review or QA repeatedly, and what is that costing the team
- ✦ Whether productivity differences between developers are due to complexity or a workflow bottleneck, nobody's flagged

These are the numbers that turn a vague "the team feels slower lately" into a specific, fixable observation. Time in Status Reports surfaces all of them directly inside Jira.

How It Works for Engineering Managers

01 Jira Cycle Time by Assignee: See Where Time Actually Goes, Per Developer

Time in Status Reports breaks down how long each issue spends with a given assignee, status by status, not just a single cycle time number for the whole team. This means you can see whether a developer's tickets are slow because of coding time, or because they're sitting in "Waiting for Review" while someone else is the bottleneck.

Use case: A developer's tickets look slower than the team average. Cycle Time by Assignee shows their code review time is actually faster than most, the delay is happening in QA handoff, not in their own work. You fix the right problem instead of the wrong one.

02 Jira Developer Productivity Metrics: Separate Real Delays from Rework

Beyond raw duration, Time in Status Reports tracks how many times an issue re-enters a status, like bouncing back into "In Progress" after a failed review. Repeated re-entries are a signal, not noise: they usually point to unclear requirements, incomplete testing, or a review process that's catching problems too late.

Use case: You notice a handful of tickets went back into "In Progress" three or four times each. Time in Status flags this rework pattern by issue and by developer, so you can address it as a process gap in code review criteria, not a performance issue with the individual.

03 **Jira Cycle Time Trend Report: Track Delivery Health Over Time, Not Just This Sprint**

A single sprint's cycle time doesn't tell you much on its own. Time in Status Reports lets you compare cycle time across weeks or months, so a gradual slowdown in "In Review" or "In Progress" doesn't stay hidden until it's already caused a missed release.

Use case: Leadership asks why the team's velocity has felt inconsistent over the last quarter. You pull the Cycle Time Trend Report, filtered by issue type, and show a clear month-over-month increase in review-stage duration, along with the sprint it started. That's a data-backed answer instead of a guess.

Get Started

Time in Status Reports is available on the Atlassian Marketplace as a Jira plugin by RVS Softek.

If you're an engineering manager who's tired of explaining delivery slowdowns after the fact, it's worth seeing what visibility into cycle time actually looks like.

[View on Atlassian Marketplace](#)[Book a Demo](#)