

## How do you calculate commit complexity?

### What is a commit complexity?

On a large team, picking which commits to review can be pretty arbitrary. Even with diligent, daily code review, it's very difficult to know which of the 100+ commits should be selected for review.

Commit complexity is a measure of how likely it is a particular commit will cause problems. Think of this as a pattern-matching engine, where Flow is looking for anomalies that might cause problems.

Here are some of the questions we ask when looking at **commit complexity**:

- How big is this commit?
- Are the changes tightly grouped or spread throughout the codebase?
- How serious are the edits being made — are they trivial edits or deeper, more severe changes to existing code?

The Flow commit complexity metric allows team leads to identify and review the most anomalous work first which, maximizes their most precious resources: time and attention. Meaning that the 30 minutes spent reviewing yesterday's work can be more than just a formality: it's a half hour spent specifically on outlier review, netting out to another pair of eyes and high-quality feedback where it's needed most.

Even on modestly-sized engineering teams, it's very difficult to prioritize specific work to review or give feedback on. Commit complexity helps teams put their attention where it's most needed. This not only helps with quality control, but serves as an incredible tool for building engineering talent: by concentrating review time on outlier commits, engineers receive high-quality feedback and suggestions for improvement where it's needed most.

[back to top](#)

---

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