How do you calculate commit complexity?

On a large team, picking which commits to review is often arbitrary. Even with diligent, daily code review, it’s very difficult to know which of your commits should be selected for review.

Who can use this?

Core

✓

Plus

Commit complexity is a measure of how likely it is that 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. Identifying commits with higher Commit complexity lets you focus your review attention where you need it most—making reviews not just a formality but a space for high-quality feedback.

Commit complexity is 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.

Tip: Commit complexity and Impact can seem very similar. The two metrics serve two different purposes and are not interchangeable. While Commit complexity looks at the riskiness of a commit, Impact considers the cognitive load of the work being done.

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