FAQ

This is a list of questions / frequently occurring issues when working with code reviews and answers how you can possibly tackle them.

What Makes a Code Review Different from a PR?

A pull request (PR) is a way to notify a task is finished and ready to be merged into the main working branch (source of truth). A code review is having someone go over the code in a PR and validate it before it is merged, but, in general, code reviews can take place outside PRs too.

Code Review	**Pull Request**
Source code focused	Intended to enhance and enable code reviews. Includes both source code but can have a broader scope (e.g., docs, integration tests, compiles)
Intended for **early feedback** before submitting a PR	Not intended for **early feedback**. Created when author is ready to merge
Usually a synchronous review with faster feedback cycles (draft PRs as an exception). Examples: scheduled meetings, over-the-shoulder review, pair programming	Usually a tool assisted asynchronous review but can be elevated to a synchronous meeting when needed

Why do we Need Code Reviews?

Our peer code reviews are structured around best practices, to find specific kinds of errors. Much like you would still run a linter over mobbed code, you would still ask someone to make the last pass to make sure the code conforms to expected standards and avoids common pitfalls.

PRs are Too Large, How can we Fix This?

Make sure you size the work items into small clear chunks, so the reviewer will be able to understand the code on their own. The team is instructed to commit early, before the full product backlog item / user story is complete, but rather when an individual item is done. If the work would result in an incomplete feature, make sure it can be turned off, until the full feature is delivered. More information can be found in Pull Requests - Size Guidance.

How can we Expedite Code Reviews?

Slow code reviews might cause delays in delivering features and cause frustration amongst team members.

Possible Actions you can Take

- Add a rule for PR turnaround time to your work agreement.
- Set up a slot after the standup to go through pending PRs and assign the ones that are inactive.
- Dedicate a PR review manager who will be responsible to keep things flowing by assigning or notifying people when PR got stale.
- Use tools to better indicate stale reviews - Customize ADO - Task Boards.

Which Tools can I use to Review a Complex PR?

Checkout the Tools for help on how to perform reviews out of Visual Studio or Visual Studio Code.

How can we Enforce the Code Review Policies?

By configuring Branch Policies , you can easily enforce code reviews rules.

We Pair or Mob. How Should This Reflect in our Code Reviews?

There are two ways to perform a code review:

1. Pair - Someone outside the pair should perform the code review. One of the other major benefits of code reviews is spreading knowledge about the code base to other members of the team that don't usually work in the part of the codebase under review.

2. Mob - A member of the mob who spent less (or no) time at the keyboard should perform the code review.

Last update: August 26, 2024