Code reviews and pull requests

Pull request

Recall that a pull request (PR) is when a feature branch is being merged into the main development branch.

The main development branch should always be in a working state! It is usually the deployed version of the code.

Don't break the build

Feature branches should have working tests.

Usually tests will be run automatically on push.

If you merge the branch and it causes the main branch to fail, you've broken the build!

Qualities of a good PR

Tests must be passing.

Code is readable (well formatted and organized).

System architecture is well designed.

Code meets the requirements and specifications of the feature it implements.

Code doesn't cause changes in behavior in other parts of the code base.

Change course early

Don't wait until you are feature complete to get feedback from your peers. If your design is fundamentally flawed, you should figure it out long before you spent a week implementing it.

Be respectful

We put a lot of ourselves into our work. If someone critiques our work, it can feel like a personal attack.

Feedback should discuss the code, not the skills or actions of the person involved.

Prefer to ask questions rather than make statements.

Don't make snap judgements or assumptions. Interrogate your reaction and ask for clarification.

Focus the critique on yourself rather than the person (e.g. I had a hard time understanding this, could you explain it?).

Don't focus on trivialities like style.

Give suggestions & explanations rather than commands.

You have a bug here that needs to be fixed.

This code is a mess and poorly written.

This is confusing.

Rename this variable.

This won't work. The design is fragile and will be hard to change.

You didn't follow the correct patterns established in the code base.

You didn't follow the style guide. You need to indent the code. These braces are in the wrong place.

This will fail if someone enters in a negative number.

If we keep writing code like this, no wonder the code base is so bad.