

# Code Review Checklist

## Before opening a code review

- ☐ (AUTHOR) All relevant requirements are met (and/or all defects are corrected)
- ☐ (AUTHOR) All relevant user/API/README documentation is updated
- ☐ (AUTHOR) All automated tests pass
- ☐ (AUTHOR) Code has been developer-tested to the extent it is testable
- ☐ (AUTHOR) Completed a full, final self-review of the code, checking for the same things as a peer reviewer
- ☐ (AUTHOR) The code builds without error or warning

## During the code review

- ☐ (AUTHOR) The review is given a clear introductory description that explains the issue or feature being solved, links to relevant external documents, and gives the context of the solution that might help reviewers better understand the code
- ☐ (AUTHOR) Comments are placed throughout the review tool to help explain the motivations behind certain changes, important trade offs or alternatives considered, or to highlight areas of particular concern that should be deeply reviewed.
- ☐ All relevant requirements are met (and/or all defects are corrected)
- ☐ The code adequately addresses all use cases and edge cases
- ☐ The code appropriately employs defensive coding and sanitizes data from users or APIs
- ☐ User interfaces match the visual designs
- ☐ All code whose purpose is not readily obvious has been appropriately commented
- ☐ All leftovers from abandoned approaches have been cleaned out (including code that is commented-out)
- ☐ No files are missing from the review
- ☐ The code is coherent and well structured, and updated sections follow local conventions and otherwise fit into their larger context
- ☐ Code conforms to the team's coding standards (formatting, conventions, naming, etc.)
- ☐ All relevant user/API/README documentation has been updated clearly
- ☐ All automated tests and linters pass
- ☐ Non-functional requirements are properly considered. These include but are not limited to the following:
  - ☐ Performance
  - ☐ Security
  - ☐ Accessibility
  - ☐ Responsive Design
  - ☐ Progressive Enhancement

- ☐ Browser Compatibility
- ☐ The implementation does not break the overall architecture of the application (e.g. inefficient code in what needs to be a performant subsystem, calling a subsystem that should not be referenced, etc.)

## After the code review

- ☐ (AUTHOR) Add unresolved issues from the code review into the issue tracking system
- ☐ (AUTHOR) Make sure the code is merged into the appropriate branch
- ☐ (AUTHOR) Update the status of issues in your ticket tracking system