Code Review Checklist

Contents

This checklist is a guide for reviewing code changes. It can be used as a reference for both authors and reviewers to ensure that the code meets the project's standards and requirements.

Code Quality

- Is the code clear and understandable?
- Does the code follow the project's coding conventions and style guide (naming conventions, spacing, indentation, etc.)?
- Are there any redundant or unnecessary parts of the code?
- Is there duplicated code that could be refactored into a reusable function/method?
- Are there any magic numbers or strings that should be constants or configurations?

Architecture and Design

- Is the code change consistent with the overall architecture of the system?
- Are the classes, modules, and functions well-organized and appropriately sized?
- Are design patterns used appropriately and consistently?
- Does the change introduce any potential scalability issues?
- Is there a clear separation of concerns (e.g., UI, business logic, data access)?

Functionality

- Does the code do what it's supposed to do?
- Are all edge cases considered and handled?
- Is there any dead or commented-out code that should be removed?
- Are there any debugging or logging statements that need to be removed or adjusted?

Security

- Are all data inputs validated and sanitized to prevent SQL injection, XSS, etc.?
- Are passwords and sensitive data properly encrypted or secured?
- Are there any potential security vulnerabilities introduced or exposed by the code change?
- Is authentication and authorization handled properly?

Performance

- Are there any obvious performance issues or bottlenecks?
- Is the code optimized for time and space complexity where necessary?
- Are large data sets or files handled efficiently?
- Is caching implemented appropriately?

Testing

- Are there unit tests covering the new functionality or changes?

- Do the existing tests need to be updated or extended?
- Is there appropriate error handling and logging in the tests?
- Do all tests pass?
- Is there enough coverage for critical paths in the code?

Documentation and Comments

- Is the new code adequately commented for clarity?
- Is the documentation (README, API docs, inline comments) updated to reflect the changes?
- Are complex algorithms or decisions well-explained?
- Are there any assumptions or limitations that need to be documented?

Compatibility

- Is the code compatible with all targeted environments (operating systems, browsers, devices)?
- Does the change maintain backward compatibility or is a migration path provided?
- Are there any dependencies added or updated? If so, are they necessary and properly vetted?

Reviewer's General Feedback

- Provide any general feedback or suggestions for improvements.
- Highlight any areas of excellence or particularly clever solutions.

Previous
Contribution Guidelines

Next
Awards and Recognition