La Vité

Vilen Milner
Michael Rossinski
Qi Hang Yang
Tianyi Zeng
Arnob Talukder

Code Review Strategy

When conducting the code review, it is important that we cover every major aspect of our code. Doing so will allow us to catch big errors that require large-scale fixing. We must also investigate our code line-by-line in order to catch small errors that could potentially be messing up the entire product. Overall, this strategy should go over all of the key points of a code review and will allow us to properly enhance our product.

Readability and Maintainability

Documentation: Group members should specifically pay attention to the internal documentation that explains what specific blocks of code do and how to add on to them. Members should identify any spelling and grammar mistakes, as well as any comments that can be reworded in a better way.

Coding Style/Naming Conventions: Group members should focus highly on how the code is presented, mainly on how easy it is to read and interpret. Duplicate lines of code must be identified, as well as lines of code that are identical after being copy/pasted. In addition, naming conventions are extremely important for the product that we are trying to implement. It is imperative that members go over every single field line and make sure that they all follow the same naming convention.

Design

SOLID Principles: When reviewing the overall design of the code, group members should recall the SOLID principles and determine whether or not they are properly followed.

Code Structure: Group members should constantly strive to find ways to clean up the code and shorten lines in order to reduce clutterness. It is important to identify areas that can be improved using design principles in order to make the code logical to others that will view it.

Functionality

Understandability: When viewing the code at any given time, group members should ask themselves whether or not they would understand what it does if they weren't the ones that implemented it. This is one of the most important aspects of the code review.

Proper Functionality: Group members should focus on making sure that the code works properly, without any errors and bugs. Any unneeded lines of code that aren't a part of the functionality should be identified. Members should also strive to find any errors or exceptions that are *not* properly handled by the system.

Testing

Understandability: Group members should make sure that the test cases are clear in their functionality, and make sense to use. Redundant and/or duplicate test cases should be identified. The results of these tests should also be displayed clearly.

Coverage: Group members should review all test cases and make sure that they have complete coverage. Edge cases must be accounted for, and any error/exception cases must also be covered.