Code Quality Metrics

This document empowers developers with an understanding of code quality metrics. By considering these metrics when reviewing finished code, developers ensure that the code is of a high standard and that other team members can easily understand what the code is doing.

Readability

This is arguably the most critical code quality metric that software developers must adhere to as it allows for developers' levels of understanding to be higher. Code created by developers must be easy to read, which allows other developers to assist in increasing the performance and quality of code. The code structure is also more consistent and relieves the debugging process.

Reliability

Reliability of code concerns an application's ability to run without encountering errors within the code and pass tests created to test its features. A developer can determine the reliability of code by carrying out a static code analysis, which is a test that uncovers faults or issues within the code. To code a reliable codebase, a low fault count is crucial.

Testability

This code quality metric will measure how well a piece of code responds to various tests performed on it. Depending on the complexity of a piece of code, more tests may be required to uncover potential faults within the program and ensure that they are swiftly taken care of. Best practices to test the durability of does are conducting unit tests and exporting all code that is non-testable into a wrapper class.

Maintainability

Maintainability concerns itself with how easily code can be maintained long after its release and how easily changes can be made while ensuring that the risks involved in these changes are kept to a minimum. Tips to ensure maintainability is improved within code is to keep code as simple as possible, continually refactor your code throughout development, and automate the build process of an application to ensure developers are able to easily compile code.

Efficiency

Code efficiency concerns how effectively a chunk of code utilizes resources, which include time (run speed) and space (memory). The fewer resources required for a program to run, the more efficient the code, making it vital for developing scalable, responsive, and resource-conserving programs.

Documentation

Code that is well-documented allows other developers to utilize it within their applications to save time and increase collaboration between developers working on the same project. It also improves code readability and maintainability and aids in knowledge transfer between developers.