

QA (Quality Assurance)

Software

Software submitted to the project must follow the following quality points:

- All methods and classes MUST have [JavaDoc](https://www.oracle.com/technetwork/java/javase/tech/index-137868.html) (<https://www.oracle.com/technetwork/java/javase/tech/index-137868.html>) style comments, covering all parameters and return types and a simple explanation of what the method/class does.
 - Note: IntelliJ provides a template for the JavaDoc comment for the method head as soon as you type `/**` + Enter above the header.
- Code should be written to use parameters wherever possible, whether for passing user input or test double elements to ensure easier testing.
- Tests must pass in accordance with the [Testing Plan \(TestingPlan.md\)](#).
- The [changelog \(../CHANGELOG.md\)](#) should be updated to describe any new, altered, or removed functionality. The template at the bottom of the document should be used to ensure consistency.

Other Documentation

The other documentation in the document should also follow some guidelines to ensure quality.

- Documents should follow the layout in the templates, making them easy to read and compare. Particularly the meeting notes.
- The [ReadMe \(../README.md\)](#) for the software should be up to date with the master branch of the project. Meaning it contains a description of what the software currently does and how to run the software.

Ensuring QA Standards Are Met.

The project will be run with a single Project Manager, also referred to as the PM, (Conor) who will be responsible for [reviews \(ReviewPlan.md\)](#) and ensuring that the quality standards are kept to by creating the templates for the documentation and reviewing the software.

The PM will also be responsible for assigning tasks to the different team members, after taking the teams preferences and skills into account, and making it clear what is expected from each of the team members by the start of the next meeting.

Breaches in the QA standards should be noted using the GitLab issues system. Any team member should use this if they notice problems with either the code or the documentation.

Amendments

- 2019-03-01: Javadoc comments are now only required on non-private methods.
- 2019-03-12: Undo Last amendment.

