Logger requirements

Functional requirements

1.1 Must Haves

1. The logger must be able to log messages to the console (eg. the Eclipse IDE console).
2. The logger must be able to log messages to a text file.
3. The logger must be able to log messages to all available output forms at the same time (for example, it must be able to log to the console and to a text file simultaneously).

1.2 Should Haves

1. Each individual log (message) should support multiple levels of importance.
2. The logger itself should have an importance level of logging.
3. All logs that have an importance level that is higher than or equal to the importance level of the logger, should be logged.
4. All logs that have an importance level that is lower than the importance level of the logger, should not be logged.

1.3 Could Haves

1. The time at which a message is logged could be included in the (final) log message.
2. The priority of a message that is logged could be included in the (final) log message.

1.4 Won't Haves

1. The logger won't include the name of the class from which the logging method is called.

Nonfunctional requirements

1. A fully functional version of the logger will be delivered on 18-09-2015.
2. The logger will be developed using the Scrum software development methodology. (The development will take one Scrum iteration)
3. The logger will be developed using the Responsibility Driven Design technique.
4. The implementation of the logger shall have at least 75% of meaningful line test coverage (where meaningful means that the tests actually test the functionalities of the logger and for example do not just execute the methods involved)