# Learning Objective 5

Conduct Reviews and Inspections and Design and Implement Automated testing Processes

## 5.1 Code Review and Issues

The code was assessed by both VSCode and IntelliJ’s built in functionalities as IDEs, highlighting errors and providing warnings regarding incorrect or potentially malfunctioning code. This was particularly helpful at the Unit level, as any errors were quickly identified before they became problematic and were simply corrected. The code was also manually inspected for functionality and the presence of every necessary method and requirement. At the time of completion, no issues were outstanding. However, the code assessment was beneficial in pointing out problems with both the pathfinding algorithm (not tested here) and the class LngLatHandler.java (tested), preventing errors from causing bigger problems later in the development process.

## 5.2 Construct a CI Pipeline

For the PizzaDronz system, a CI pipeline may be constructed in the following way:

- Repository: Using a version control system such as GitHub, a code repository should be constructed and utilised by the system. Pushing the code to the repository should execute the CI pipeline.
- Build: The PizzaDronz system utilised the tool Maven, which created a helpful environment for working with various Java libraries and JARs, which were necessary for the project. The pipeline should automatically build and run the code when it is committed to the repository, ensuring it compiles correctly and does not contain errors that would prevent this.
- Test: When pushed, the code’s tests should automatically be run by the system, which validate the system’s functionalities and ensure there are no errors present. This includes all of the tests constructed for this project (relating to **R1**-**R6**). The pipeline should also assess the tests’ code coverage levels, to ensure that the system is sufficiently tested and no omissions are made in the code.
- Deploy: Once testing has been completed and passed, the project should be submitted to a real testing environment.

## 5.3 Test Automation

All tests that were documented within this repository could be automated at the Test stage of the CI pipeline, being automatically run whenever a commit is made to the repository. This saves time for the developer, and ensures that the code is still in a functioning condition. An improvement over the discussed tests would be to include regression testing, to ensure that any modifications or new features of the code do not prevent any old functionalities and stop the system from working as expected.

## 5.4 Demonstrate CI Pipeline

The pipeline should be capable of verifying that the code has built successfully and subsequently that all automated tests have passed, confirming that the system is working as expected.