# Introduction

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest.

The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

So, it is possible to generalize that:

* Testing is important because software bugs could be dangerous.
* The important reasons for using software testing are: security, product quality, and customer satisfaction.
* The important strategies in software engineering are: unit testing, integration testing, validation testing, and system testing.

For this project it is possible to take a closer look exactly at unit testing.

# Test Environment (SimuTests.java)

To implement the tasks that were given it was decided to:

* Use the JUnit framework, it gives the opportunity to easily implement unit test using Java
* Use the VS Code IDE that is able to run this environment.

On the Fig (1) it is shown how all modules and methods that are used in platooning can work without issues.

A picture containing text

Description automatically generated

Fig. 1 – All tests passed

To be able to simulate the behavior of the system that is more likely looks like a real-world system – it was decided to use Java Random class to add the realistic scenario.

The results of the completed tests that received random data as a parameters in functions are shown on the Fig (2).

Text

Description automatically generated

Fig. 2 – Stochastic factor

One of the requirements of this implementation – is to show defect tests. In this run it can be seen that random data affected some modules and methods to behave in the different way than expected. It is shown on the Fig. (3) – all modules failed the test.

Text

Description automatically generated with low confidence

Fig. 3 – Example of bad test run

The tests are ran to find the mistakes in the code or in behavior. So, the IDE gives the opportunity to understand what went wrong. If the test was written in a good manner, it will become obvious what went wrong in the behavior. Fig. (4) shows how truck system expected truck to be a follower, but receiver the answer that it is in fact main truck.

Text

Description automatically generated

Fig. 4 – Error validation

Parametrized test was created to run multiple inputs and receive multiple outputs an once. This is shown on Fig. (5).

Text

Description automatically generated

Fig. 5 – Parameterized test function

The software tester is always notified by the test environment what exactly went wrong in the scenario to be able to improve the system. (Fig. 6).

A picture containing text

Description automatically generated

Fig. 6 – Error notification

# Truck Simulation Class (Simu.java)

To be able to run the tests the class with a logic of the behavior in needed. Fig. (7) represents the class with methods and modules that simulate the truck being:

* Put to Turned on
* Make a selftest
* Check its status in the platoon
* Enter platoon with a role
* Execute a positioning
* Execute a speed control
* Be able to communicate with other trucks

In this case, the positioning, speed control and communication were implemented like a modules that need to be tested, as it is said in the task.

Other methods were implemented to be able to make the real-world behavior representation of the truck in the platoon.

Graphical user interface, application

Description automatically generated with medium confidence

Fig. 7 – All methods and modules

Fig. (8) gives the closer look to an example of how the truck simulation logic can be implemented to be able to run the realistic tests on it.

Text

Description automatically generated

Fig. 8 – The simulation run logic example

# Conclusion

Testing is an important part on the software development because software bugs could be dangerous. The real reasons for using software testing are: to improve security, product quality, and customer satisfaction. The unit testing is the testing strategy that can be used to test the behaviors like truck platooning.