Testing

Testing is the process undertaken to verify that a program performs against its specification. The idea of testing is to make sure that the program does what it is supposed to do.

Testing can be considered as one of the following types:

White Box Testing: Logical testing where knowledge of the code is required to ensure that the code follows the paths we expect it to given a set of conditions.

Black Box Testing: Functional testing where there is no knowledge of the code that has been written.

White Box Testing

This is the most common type of testing that we do as developers. This type of testing tries to work through all the paths of our software and every line of the code to ensure the code is correct.

This type of testing starts with valid conditions to ensure that the code correctly functions.

Then tests are performed that are invalid conditions to test that the code handles errors in a correct manner.

This type of testing requires some understanding of the code that has been written both for the valid paths and to deliberately break the code. For this reason the developer is normally the person who does the initial white box testing.

White box testing the tests are from the developers perspective and is testing how the program works.

Black box testing

This is normally where the tester is not the same person as the developer. The person testing does not understand the underlying code or structure and so the tester treats the code as a black box, its contents are hidden. The tester just gets to see the inputs and the outputs.

This is a good test as the user will often perform functions that we didn’t expect. The problem is as we program our code we are so familiar with it and unknowingly don’t do things that a user might do.

All the tests in black box testing are from a user’s perspective and are designed to test what the program does. For this reason, not all the paths through the program will be covered in black box testing.

Test Plans

Test plans are the detailed tests that we are going to perform to ensure that our code functions as we expect and want it too. We need to consider the following when designing our tests:

- In general terms, what are we testing?
- What tests will I do to test that part of my program?
- What shall I use as my input data, valid and invalid?
- What output am I expecting from each test?

To get a good test plan, then these questions need to be answered. Write down the tests on a sheet of paper so that a record of the tests can be shown. A good test plan will look something like this:

System Name: Program Name: Module Name:			Version Number: Test Author:	
Test Number	Date	Purpose / Type of Test	Inputs	Expected Outputs

As much detail as possible should be provided when writing tests to assist the tester. Obviously a simple program will not have a significant number of tests but will have a number of tests that need to be performed.

The test log is where the tester uses the above plan and indicates if the tests passed or failed. This will then show where the development needs to be focused, but will also indicate to any potential client the quality of our developed product.

The tester then uses the tests that are written down to test the software and record the results in a test log. The test log will look something like this:

System Name: Program Name: Module Name:			Version Number: Tester Name:
Test Number	Date	Actual Output	Comments

If the program is complex and has failed testing, the test plan will remain the same (although some tests may be added) and a NEW test log is created for the new round of testing. The original test log is kept as a separate document.