

Component 2: Testing and Refinement

Testing

Term	Definition
Test strategy	A test strategy is a document that sets out an approach to testing a software application
Test Plan	A test plan is a document that details the scope, approach, resources and schedule of intended test activities.
Refinement	Testing results are used to refine / improve the functionality of a program.

Test strategy

An effective testing strategy will ensure that a final solution to a problem meets the original requirements of the user. The strategy will describe how the outcomes of the testing process can be used to inform further development of the solution.

Unit testing

Unit testing is a testing technique that involves testing each individual module of a programmed solution to ensure that each one functions as it should.

Integration testing

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

System Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.

Acceptance testing

Acceptance testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the user's requirements and assess whether it is acceptable for implementation in the user's organisation.

Test data

The design of test data is crucial to the effectiveness and success of the testing process. Test data should include examples of typical, extreme and erroneous data.

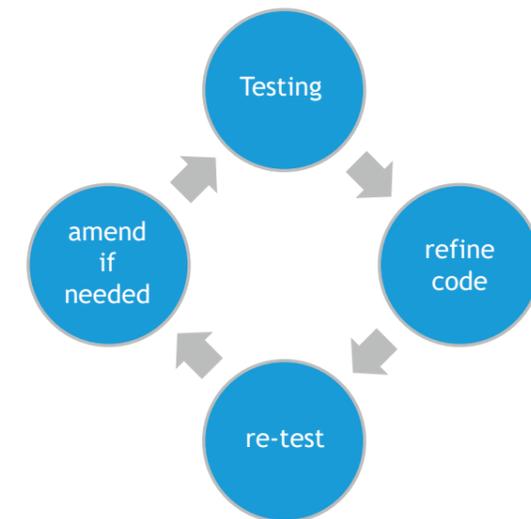
Typical / standard data - data that is correct and should be processed correctly by the program.

Extreme / boundary data - data that is correct but is at the extreme of a boundary of an acceptable range of values.

Erroneous data - data that would cause the program to fail if not validated and rejected.

Refinement

To be able to refine a program after testing having been completed you need to have a complete understanding of the original requirements for the program and the changes needed to meet any revised requirements that have been identified during the development process.



To carry out the refinement of a program you need to:

- Understand the strengths and weaknesses of the program you are refining
- Evaluate the outcomes of the original code
- Be able to evaluate the outcomes of the changes you have made
- Address any additional requirements of the end user of the program following the testing outcomes.