Black Box Testing - Validation

Testing Principle

It is good practice to carry out testing such that the test harness does not interact with or affect the software being tested. The test harness should treat the software being tested as a 'Black Box' which takes input values and returns output values. The test results so obtained show the innate response of the Black Box of software to its inputs. They show its behaviour without regard to the nature of its construction. (This does not necessarily mean that no regard is paid to the nature of its construction in devising the tests).

Once the tests have shown the software to be valid, then it can be plugged into the Raquel DBMS. This is important because the Raquel DBMS is designed to consist of a set of modules plugged together, that interact only via their input and output interfaces.

Black Box testing is used to ***validate*** the software; i.e. to make sure that it behaves correctly.

Because Black Box testing should be carried out at every level from small units to the whole system, it also helps to ***verify*** the software; i.e. together with the correct design algorithms and software construction techniques, it makes sure that the software is built correctly.

The Test Harness

The test harness should provide 2 categories of facilities :

1. Those that provide a test environment. They provide a means of inputting data to the software being tested, receiving output from it, and handling any error conditions that it may generate.
2. Debugging tools that interact with the software being tested via 'hooks' provided by the compilation process. The source software is unchanged but the executable code produced from it includes functionality not otherwise present, so that debugging tools can use that functionality to trace the software's execution and inspect its internals at various points. This relies on the quality of the debugging tools to maintain the 'Black Box' effect.