Test Case

According to my study about the test cases, a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Cases should also be considered that are based on information sources such as risks, constraints, technologies, change requests (defects), faults and so on.

So we run tests in order to:

**Find defects**

**Maximize bug count**

**Minimize technical support costs**

**Assess conformance to specification**

**Conform to regulations*.***

If a regulation specifies a certain type of coverage (such as, at least one test for every claim made about the product), the test group creates the appropriate tests. If the regulation specifies a style for the specifications or other documentation, the test group probably checks the style. In general, the test group is focusing on anything covered by regulation and (in the context of *this objective*) nothing that is not covered by regulation.

**Find safe scenarios for use of the product (find ways to get it to work, in spite of the bugs**

**)**

**Verify correctness of the product**

The important information about the test cases:

Additional information that may be included:

* test case ID
* test case description
* test step or order of execution number
* related requirement(s)
* depth
* test category
* author
* Check boxes for whether the test is automatable and has been automated.

Additional fields that may be included and completed when the tests are executed:

* pass/fail
* remarks

Larger test cases may also contain prerequisite states or steps, and descriptions.

**Checkpoints: Test Case**

* + A description of the use case, use-case scenario, test objective, or condition being evaluated has been clearly stated for each test case.
  + Each test case states the expected result and method of evaluating the result.
  + For each requirement for test, at least two test cases have been identified. One test case, representing an expected condition, is developed to verify the correct or expected behavior (positive test). A second test case, representing an unacceptable, abnormal, or unexpected condition, is developed to verify the requirement for test does not execute in an unexpected manner (negative test). Typically, for each requirement for test there will be at least one positive test case and many negative test cases.
  + Test cases have been identified to execute all product requirement behaviors in the target-of-test, including (as appropriate):
    - function
    - data validation
    - business rules implementation
    - target-of-test workflow or control
    - dataflow
    - object state
    - performance (including workload, configuration, and stress)
    - security and accessibility
    - compliance
  + Each test case describes or represents a unique set of inputs or sequence of events that results in a unique behavior by the target-of-test. Review those test cases that produce the same behavior and determine if they are equivalent, that is, they both execute the path in the target-of-test.
  + Each test case, or group of related test cases, identifies the initial target-of-test state and the state of the test data.
  + All test case names and/or IDs are consistent with the test artifact naming convention.

Test cases are the single most important element of the test plan. The test case defines the what, where, why and how of a test. Specifically, what is to be tested, on which platforms it is to be tested, the purpose of the test and how the test will be executed. In terms of execution, Rational Quality Manager has the native ability to author and execute manual test scripts as well as the ability to launch automated test scripts from Rational Functional Tester, Rational Performance Tester and Rational Service Tester. Various execution adapters are also available which enable execution of other tests from third party vendors.