**Best Practices To Write Good Test Case**

Characteristics of a good test case are

* Easy to understand and execute
* Create Test Cases with End User’s perspective
* Unique Test case Identifiers must be used. It allows us to track them easily.
* Prerequisites should be listed clearly. Helps to execute the test case without any issues.
* Test data should be defined to evaluate each functional area.
* Test case description should be concise.
* Test Steps should be in detail and clear.
* Specify the exact expected result.
* Position condition should be listed if any.
* Test cases should neither too simple nor too complex.
* Test cases must be distinctive. There should not be no repeated test cases.
* Test cases should be written by following [test case design technqiues](https://www.softwaretestingmaterial.com/black-box-test-design-techniques/).
* Test cases must be comprehensible. So that any tester (even a newly appointed testers) can understand them by perusing once.
* Needs to provide clear environment details where we need to execute them.
* Test cases should be reusable & maintainable
* Get peer review.

If you follow the best practices to write test cases then anyone in the team can understand and execute the well-written test case easily. It should be easy to read and understand, not only for whoever wrote it but also for other testers as well.

**Easy to understand and execute:**

To make the test cases easy to understand and execute faster we need to use simple and easy to understand language like “Go to login page”, “enter username”, “enter password”, “click on login button” and so on.

**Create Test Cases with End User’s perspective:**

Create test cases by keeping end-user in mind. The test cases you create must meet customer requirements.

**Use unique Test Case ID:**

It’s good practice to follow a unique id with some naming convention for better understanding and discrimination purposes.

**Have a clear description:**

Your test case description should be clear enough to understand what the tester is going to do with this test case.

**Add proper preconditions & postconditions:**

In some cases, test cases need to meet some conditions before execution or achieve some conditions after execution. These conditions we need to mention properly in the Pre and postconditions.

**Specify the exact expected result:**

Include the Expected result: Expected result tells us what will be the result of a particular test step. Testers decide the pass or fail criteria based on the expected result.

**Test cases should be reusable & maintainable:**

A well written test case is reusable and maintainable. There ar times where developers change the code, and testers need to update the test cases. If our test cases are easy to read and understand then it would be easy to update them not only by whoever wrote it but also by other testers as well.

**Utilize testing techniques:**

Use testing techniques whenever necessary. Software test design techniques are used to derive the test cases from the Requirement Specification document and also based on testers expertise

* [Equivalence Partitioning](https://www.softwaretestingmaterial.com/equivalence-partitioning-testing-technique/)
* [Boundary Value Analysis](https://www.softwaretestingmaterial.com/boundary-value-analysis-testing-technique)
* [Decision Table](https://www.softwaretestingmaterial.com/decision-table-test-design-technique/)
* [State Transition](https://www.softwaretestingmaterial.com/state-transition-test-design-technique/)
* [Exploratory Testing](https://www.softwaretestingmaterial.com/exploratory-testing-tutorial/)
* Error Guessing

**Get peer review:**

Let your peers review your test cases to uncover the defects in your test case design.

**Popular Test Case Management Tools**

some of the popular tools for the test management process are

1. PractiTest
2. Test Rail
3. Testpad
4. Qase
5. Klaros
6. Test Collab
7. QMetry
8. Meliora Testlab
9. TestLodge
10. TestCaseLab