

Software Testing Principles

A principle can be defined as:

- a general or fundamental, law, doctrine, or assumption;
- a rule or code of conduct;
- The laws or facts of nature underlying the working of an artificial device.



Principle 1/11

Testing is the process of exercising a software component using a selected set of test cases, with the intent of (i) revealing defects, and (ii) evaluating quality

- Execution-based activity to detect defects.
- Separation of testing from debugging since the intent of the latter is to locate defects and repair the software.



Principle 2/11

When the test objective is to detect defects, then a good test case is one that has a high probability of revealing a yet undetected defect(s).

- The goal for the test is to prove/disprove the hypothesis, that is, determine if the specific defect is present/ absent.
- Based on the hypothesis, test inputs are selected, correct outputs are determined, and the test is run.
- Results are analyzed to prove/disprove the hypothesis.



Principle 3/11

Test results should be inspected meticulously

- Testers need to carefully inspect and interpret test results.
- Several erroneous and costly scenarios may occur if care is not taken



Principle 4/11

A test case must contain the expected output or result

- Expected outputs allow the tester to determine
 - whether a defect has been revealed, and
 - Pass/ fail status for the test.



Principle 5/11

Test cases should be developed for both valid and invalid input conditions.

- A tester must not assume that the software under test will always be provided with valid inputs.
- Software users often make typographical errors even when complete/correct information is available.
- Invalid inputs also help developers and testers evaluate the robustness of the software, that is, its ability to recover when unexpected events occur



Principle 6/11

The probability of the existence of additional defects in a software component is proportional to the number of defects already detected in that component

- The higher the number of defects already detected in a component, the more likely it is to have additional defects when it undergoes further testing.



Principle 7/11

Testing should be carried out by a group that is independent of the Development group

- It is difficult for a developer to admit or conceive that software he/she has created and developed can be faulty.
- Testers must realize that (i) developers have a great deal of pride in their work, and (ii) on a practical level it may be difficult for them to conceptualize where defects could be found.
- Independence of the testing group does not call for an adversarial relationship between developers and testers.



Principle 8/11

Tests must be repeatable and reusable.

- It is also useful for tests that need to be repeated after defect repair.
- The repetition and reuse of tests is also necessary during regression test (the retesting of software that has been modified) in the case of a new release of the software.



Principle 9/11

Testing should be planned.

- Test plans should be developed for each level of testing, and objectives for each level should be described in the associated plan.
- Careful test planning avoids wasteful throwaway tests and unproductive and unplanned test-patch-retest cycles that often lead to poor-quality software and the inability to deliver software on time and within budget.



Principle 10/11

Testing activities should be integrated into the software life cycle.

- It is no longer feasible to postpone testing activities until after the code has been written.
- Organizations can use process models like the V-model or any others that support the integration of test activities into the software life cycle.



Principle 11/11

Testing is a creative and challenging task.

- Creative
- Face difficulties