<https://www.softwaretestinghelp.com/functional-testing-interview-questions/>

**Q #1) What do you understand by the term ‘Functional testing’?**

**Answer:** A black box testing technique, where the functionality of an application is tested to generate the desired output by providing certain input is called ‘Functional testing’.

The role of functional testing is not only to validate the behavior of the application as per the requirement document specification but is also to verify whether the application is ready to be released into the live environment or not.

**Given below are few functional testing techniques that are commonly used:**

* Unit testing
* Smoke testing
* Integration testing
* System Testing
* Usability testing
* Regression testing
* User Acceptance testi

**Q #2) What are the important steps that are covered in Functional testing?**

**Answer:** **Following are the steps that should be covered as a part of functional testing:**

* Understanding the Requirement document specification and clearing the doubts and queries in the form of review comments.
* Writing the test cases with respect to the requirement specification by keeping in mind all the scenarios that should be considered for all the cases.
* Identifying the test inputs and requesting the test data that is required to execute the test cases as well as to check the functionality of the application.
* Determine the actual outcomes as per the input values to be tested.
* Execute the test cases that determine whether application behavior is as expected or any defect has occurred.
* Compare the actual result and the computed result to find out the actual outcome.

**Q #3) Explain the difference between Functional testing and Non-Functional testing.**

**Answer:** The difference between Functional testing and Non-functional testing can be found in the table given below:

| **Functional Testing** | **NonFunctional Testing** |
| --- | --- |
| Functional testing is performed to determine the system behaviour as per the client functional requirements. | Non-functional testing is the process to determine the system performance as per client expectations |
| Functional testing is performed first with the help of Manual and Automation testing tools. | Non-functional testing is performed after functional testing with the effective tools required. |
| It is easy to perform manual testing as client requirements are the input in functional testing. | It is difficult to perform manual testing as scalability, reliability, speed and other performance parameters are input in non functional testing. |
| Functional testing is of following types: • Unit Testing • Smoke Testing • Sanity Testing • Integration testing • User Acceptance testing • Regression testing | Non-functional testing is of following types: • Performance testing • Load, Stress, Volume Testing • Security testing • Compatibility testing |

**Q #4) How is ‘Build’ different from ‘Release’?**

**Answer: Build**is an executable file which refers to that part of an application which is handed over to a tester to test the implemented functionality of the application along with some bug fixes. The build can be rejected by the testing team if it does not pass the critical checklist which contains the major functionality of the application.

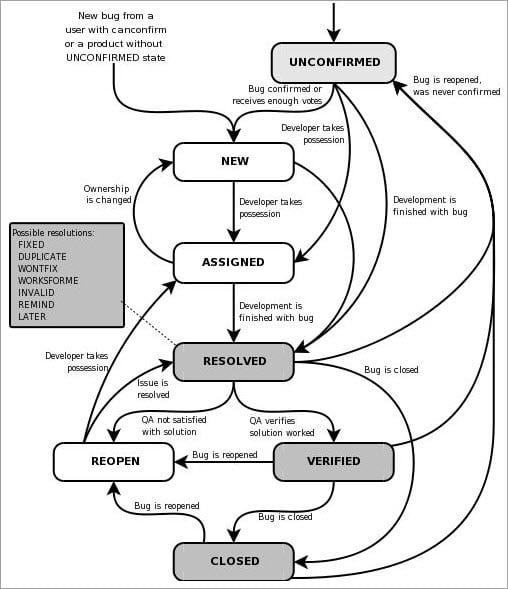
There can be multiple builds in the testing cycle of an application.

**Release** refers to the software application which is no longer in the testing phase and after completion of testing and development, the application is handed over to the client. One release has several builds associated with it.

**Q #5) Explain Bug cycle.**

**Answer:** Bug is said to be an unwanted error, flaw, mistake, etc that has occurred within the application and prevents it from delivering the desired output. When any defect or bug is encountered in an application while testing, then from logging a defect till its resolution, a bug moves through a definite life cycle known as Bug Lifecycle.

**Below figure will give you an idea of Bug lifecycle:**

[](https://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2018/01/Bug-Lifecycle.jpg)

***image source: Bugzilla bug life cycle***

The whole process goes as and when an issue or bug is encountered. It is reported /logged in bug tracking tool following a considerable format. These bugs are assigned to the developer and its status is made as ‘Open’. Developer can now review the bug, reproduce it at its end and start working on it.

If the bug is fixed, the developer changes its status to ‘Fixed’ or the status can be moved to ‘need more information’, ‘won’t fix’, ‘cannot reproduce’ etc., in other cases. QA then performs regression i.e. re-verify the bugs with a specific action and responds accordingly.

If the issues/bug is now behaving as expected then its status is changed to Verified /Closed else Reopen.

**Q #6) Enlist some Bug status along with its description.**

**Answer:** **Enlisted below are few bug statuses along with their descriptions:**

* **New:** When the defect or bug is logged for the first time it is said as New.
* **Assigned:** After the tester has logged a bug, his bug is being reviewed by the tester lead and then it is assigned to the corresponding developer team.
* **Open:** Tester logs a bug in the Open state and it remains in the open state until the developer has performed some task on that bug.
* **Resolved/Fixed:** When a developer has resolved the bug, i.e. now the application is producing the desired output for a particular issue, then the developer changes its status to resolved/fixed.
* **Verified/Closed:** When a developer has changed the status to resolved/fixed then the tester now tests the issue at its end and if it’s fixed then he changes the status of the bug to ‘Verified/Close’.
* **Reopen:** If a tester is able to reproduce the bug again i.e. the bug still exists even after fixing by the developer, it’s status is marked as reopen.
* **Not a bug/Invalid:** A bug can be marked as invalid or not a bug by the developer when the reported issue is as per the functionality but is logged due to misinterpretation.
* **Deferred:** Usually when the bug is of minimal priority for the release and if there is lack of time, in that case, those minimal priority bugs are deferred to the next release.
* **Cannot Reproduce:** If the developer is unable to reproduce the bug at its end by following the steps as mentioned in the issue.

**Q #7) What is known as Data-driven testing?**

**Answer:** Data-driven testing is the methodology where a series of test script containing test cases are executed repeatedly using data sources like Excel spreadsheet, XML file, CSV file, SQL database for input values and the actual output is compared to the expected one in the verification process.

**For Example:** Test studio is used for data-driven testing.

**Some advantages of data-driven testing are:**

* Reusability.
* Repeatability.
* Test data separation from test logic.
* The number of test cases is reduced.

**Q #8) What are the important points that should be considered while writing Test Cases?**

**Answer:** Writing a test case is said to be the most important activity of the test execution process which requires writing skills as well as in-depth knowledge of the application to make effective and reusable test cases.

**Few important points that should be considered while writing test cases includes:**

* There should be a clear understanding of the client’s requirements before beginning to write the test cases. Nothing should be assumed and every doubt regarding the requirements should be cleared.
* Every requirement should be included in the form of test cases and nothing should be left out. Usually Traceability matrix is maintained to keep a check on every requirement implementation and testing completion.
* As per the requirement document specifications, every functional and non-functional requirement including UI interface, compatibility should be covered.
* Test cases should be checked from time to time for no repetition or redundancy.
* Priority is an important factor which should be set for the test cases while writing. This priority helps tester to test the application first with the high priority tests cases which includes basic functionality, then the medium and later the low priority test cases.
* For a particular release, test cases can also be built Sprint wise so that the tester, as well as the developer, can analyze the quality of the product based on test case execution.
* Structure of test cases should be easily understood and must be in a simple language. The input data values for test cases should be valid as well as in a wide range

**Q #9) What is Automation testing?**

**Answer:** Automation testing is a testing methodology where automation tool is used to execute the test cases suite in order to increase test coverage as well speed to test execution. Automation testing does not require any human intervention as it executes pre-scripted tests and is capable of reporting and comparing outcomes with previous test runs.

Repeatability, ease of use, accuracy, and greater consistency are some of the advantages of Automation testing.

**Some automation testing tools are listed below:**

* Selenium
* Tellurium
* Watir
* SoapUI

**Q #16) What is Adhoc testing?**

**Answer:** Adhoc testing, usually known as random testing is a form of testing which does not follow any test case or requirement of the application. Adhoc testing is basically an unplanned activity where any part of the application is randomly checked to find defects.

In such cases, the defects encountered are very difficult to reproduce as no planned test cases are followed. Adhoc testing is usually performed when there is a limited time to perform elaborative testing.

**Q #20) When do we perform Smoke testing?**

**Answer:** Smoke testing is performed on the application after receiving the build. Tester usually tests for the critical path and not the functionality in deep to make sure, whether the build is to be accepted for further testing or to be rejected in case of broken application.

A smoke checklist usually contains the critical path of the application without which an application is blocked.

<https://www.softwaretestinghelp.com/how-to-write-test-cases-for-a-given-scenario/>