functional testing

Functional tests verify each function of the application or software. The tester verifies the function using a specified set of requirements. Therefore, in this case, the source code of the software or application does not play a major role. The behavior of testing software is the main concern.

Different types of functional tests include:

Unit test. In unit testing, testers examine individual software components. The purpose is to test whether the behavior of the component meets the requirements.

Integration testing. Integration testing involves testing a single component or module after it is merged into a group.

System test. Here, testers execute test cases to verify the compliance and specifications of integrated and completed software.

Intelligence test. This will test the logical reasoning related to the program's work.

Smoke test. Smoke test tests simple and basic functions, such as whether users can log in or log out.

Interface test. These tests check that the communication between the two software systems is correct.

Regression test. This is probably one of the most important test phases. Here, the old test cases for the entire application are executed after the new functionality is implemented.

Beta / acceptance test. Here, the target user tries the product and reports an error.

Non functional test

Non functional tests consider parameters such as reliability, availability and performance. A non functional test may be to check how many users can log in to the system at the same time.

Non functional test types include:

Performance test. The performance or speed of the application is tested under the required workload.

Load test. This will test the behavior of the application under heavy workload. Therefore, if you are testing a website, the load test will check the function and performance of the website under high traffic.

Pressure test. Stress testing determines the robustness of the software by evaluating whether the software is outside the normal operating range.

Batch testing. This tests the performance of the system by loading the database into the increased amount of data.

Safety test. Here, test cases are executed to check whether the system is suddenly or deliberately attacked from internal and external sources.

Compatibility test. Test cases are executed to check whether the application is compatible with different environments. For example, if you are testing a web application, the compatibility test will deal with how the web site works on different browsers or devices.

Installation test. These tests check that the product works as expected after installation.

Resume the test. Here, the tester determines the ability of the application to recover from hardware crashes and failures.

Reliability test. This procedure examines where an application can perform a specific task within a specific time frame without failure. For example, suppose you are testing a cryptocurrency mining application. A scenario where an application can mine for eight hours without crashing may be what you're looking for during reliability testing.

Availability test. Usability testing explores the ease of use of end users in learning, operating, and preparing inputs and outputs.

Compliance testing. This determines whether the system meets external and internal standards.

Localization testing. Here, testers check the behavior of products according to local or cultural settings and environment.

According to your knowledge of test products, software testing can be divided into different types: black box testing, white box testing and gray box testing.

Black box test

In such tests, you have the least information about how the product is built. You do not know the structure, code, or logic of the product. You will use the product like an end user. Because in the black box test, you will have the same amount of information as the customer, so it is used for function test.

This type of test can only occur when code is executed. Therefore, dynamic testing is used. Dynamic testing is the type of code you must execute and test products during code execution. It is mainly to check how it will start and run, and how users will experience it.

White box test

In the white box test, you have most of the information about the product. White box testing is mainly used to make code better. In such tests, it is found that the code is inefficient, the coding practice is poor, and unnecessary lines of code. Most code optimizations and security fixes are the result of passing this test.

White box testing focuses on how web applications work. It focuses more on how to make it better. You can make many improvements to the product, but the last few steps to make it perfect are difficult. It cannot be perfect unless it has no problems.

To make it perfect, you need to check it thoroughly. Because the product being executed does not give you all the insights, you must check the code without executing it. This is called static testing.

Static testing is also used in the early stages of development. When it is simple, you don't have to wait for product deployment.

Grey box test

In such tests, you have some information about the product. This type of testing helps find errors that users don't know.

For a very simple example, if you design an element with a blue shadow, but it has a green shadow. Users will not know that this is an error because they will think it should be. But your partial knowledge of the product will help you identify such errors.

No company can underestimate the importance of providing customers with the best products. The types of tests are constantly evolving and the lists are constantly changing. Depending on the nature and scope of the product, you can run different test processes.

In automated testing, testers use scripts to test (so as to automate the process). The pre write test runs automatically to compare the actual results with the expected results. In the case of test automation, things like regression testing and performing repetitive tasks seem effortless when continuous human intervention is not required.

# End To End Testing：

End to end tests are test categories drawn at the top of the test pyramid. The test pyramid gives us a guide to how many tests we should perform. Because they are easy to write and maintain and run fast, many unit tests are valuable.

Step：

Identify test scenarios.

Map the steps in each scene.

Use the mapping steps to perform manual tests.

Test automation.

Add your test to the CI pipeline

Usability test

Usability testing is a test method to measure the ease of use of an application from the perspective of end users, usually in the system or acceptance test stage. The goal is to determine whether the visible design and aesthetics of the application meet the expected workflow of various processes, such as login applications. Usability testing is a great way for teams to review the intuitive use of independent functions or the entire system.

Advice

1.time spent analyzing audio-visual materials, finding existing problems and recommending solutions

2.time spent discussing changes and modifications with developers and writing reports on findings and recommendations.

3.Planned time: determine the main test questions, the types of users to be tested, the screening questionnaire of recruited users and the test scenario. Recruitment cost: the time of the company's personnel and the cost to the recruitment company (usually a good choice). Usability experts need to spend time getting familiar with the website and its production team and designing corresponding test scenarios. If you need to record the test process, you also need to spend the rent of the laboratory or portable recording equipment.