What is Ad hoc Testing

Adhoc testing is a type of unplanned testing which does not follow any formal process like requirement documents, test plan, test cases, etc.

It is a black-box testing method that focuses on both logical and illogical scenarios randomly. Ad Hoc testing uncovers bugs that we wouldn't have found when we followed the formal process.

It is usually done after the formal testing; this helps in finding the loopholes in the application.

It is an unstructured type of software testing.

Example of Adhoc Testing

Scenario 1: Different Browser Setting

We can tweak the browser settings and check how the application responds or renders.

Let's say we have a working web application; we can disable JavaScript in another browser and test the application there.

Scenario 2: Cross-platform Compatibility

Usually, a formal testing procedure focuses on testing an application where it receives most of its traffic.

In Adhoc, we can test the application in different platforms and devices which are usually not covered in the test cases.

Scenario 3: Validity Verification

Test case designs focus on boundary value analysis and equivalence partitioning for validation of an application.

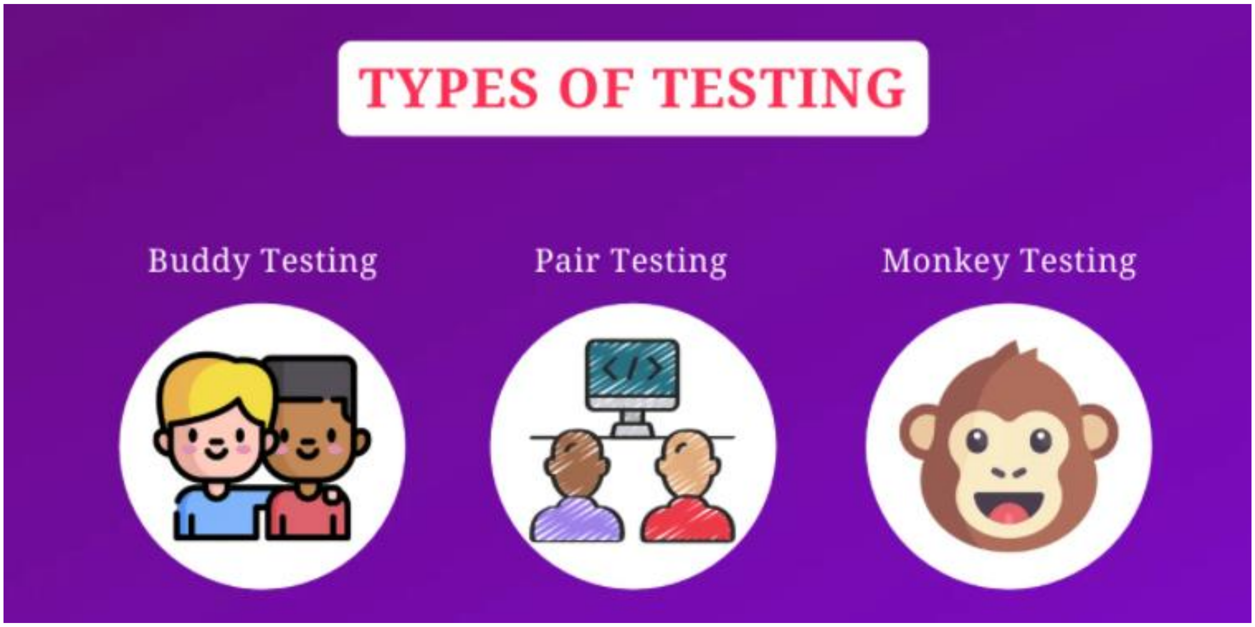
In Adhoc testing, we can throw a bunch of valid/invalid input to check how the system reacts to such data and how it handles the error.

We can provide invalid inputs to the application and check whether the application responds appropriately or not.

Importance of Ad Hoc Test

- Ad-hoc testing can find loopholes in the test strategy.
- It can improve the quality of the testing process at less cost.
- It helps testers to improve testing processes and scripted methods.
- It finds defects quickly using outside-of-the-box testing.

Types of Adhoc Testing



Even though Ad hoc testing is an unstructured type of testing we must follow certain conditions to get the best results.

Buddy Testing:

In this type of testing, at least two teammates execute the test, ideally, a developer and a tester will be testing the same module.

It usually takes place after unit testing. This helps the developer understand the testing process which in turn makes better code and it also helps the tester to avoid wasting time on reporting invalid bugs.

Pair Testing:

In this type of testing, two testers work together where one executes the test cases and the other documents.

They share ideas, opinions, and knowledge over the same product to identify errors and defects from different perspectives based on knowledge and expertise level.

Monkey Testing:

In this type of testing, a single tester performs the testing randomly intending to break the system.

It is purely based on the creativity of the tester to provide by random inputs, the output is observed.

The objective here is to find all the bugs and inconsistencies with the planned behavior.

When we perform Adhoc testing

Ad hoc testing can be really versatile, it can be done at any phase of the SDLC, it applies from unit testing to system testing and it can be done by testers, developers or any other team members i.e anyone with good knowledge about the product can perform Adhoc testing.

- It is done only when the product is stable.
- It is usually done after formal testing is completed.
- It is performed when the testing time doesn’t have much time to do exhaustive testing.
- It is performed when the team doesn’t have any task in hand, i.e during their free time or system downtime.

- Sometimes testers note down scenarios during formal testing that might have an issue and verify them when they have time, this will also be part of Adhoc testing.

As Adhoc testing can be done anywhere, anytime by anyone, it is important to know when it is not applicable.

- It should not be a part of beta testing.
- It should not be done where documentation is necessary.
- It should not be performed in test cases that already have errors in them i.e areas where the system is already failing

Adhoc testing best practices

- Testers with good expertise and in-depth knowledge about the product can make a huge difference in how the Ad Hoc test cases are executed.
- Focusing on areas that are not covered in test cases can help uncover underlying issues that would have never been found.
- Testers should focus on the most important features used by the end-user.
- Proper timelines should be put in place for effective utilization of the time.

-------------END----------

Muntasir Abdullah Mizan
muntasir.abdullah01@gmail.com