Ad hoc Testing

Ad hoc Testing is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage. Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application. Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the Software testing technique called Error Guessing.

Types of Adhoc testing

There are different types of Adhoc testing and they are listed as below:

Buddy Testing-- Two buddies mutually work on identifying defects in the same module. Mostly one buddy will be from development team and another person will be from testing team. Buddy testing helps the testers develop better test cases and development team can also make design changes early. This testing usually happens after Unit Testing completion.

Pair testing --Two testers are assigned modules, share ideas and work on the same machines to find defects. One person can execute the tests and another person can take notes on the findings. Roles of the persons can be a tester and scriber during testing.

Comparison Buddy and Pair Testing: Buddy testing is combination of unit and System Testing together with developers and testers but Pair testing is done only with the testers with different knowledge levels. (Experienced and non-experienced to share their ideas and views)

Monkey Testing-- Randomly test the product or application without test cases with a goal to break the system.

The advantage of Ad-hoc testing is to check for the completeness of testing and find more defects than planned testing. The defect catching test cases are added as additional test cases to the planned test cases.

Defect seeding/Error seeding

Error seeding and mutation testing are both error-oriented techniques and are generally, applicable to all levels of testing.

Defect seeding is a practice in which defects are intentionally inserted into a program by one group for detection by another group.

The ratio of the number of seeded defects detected to the total number of defects seeded provides a rough idea of the total number of unseeded defects that have been detected.

Defect seeding is done to find capabilities of a testing team. A testing team will add few defects in the application and the other one has to find out all the defects in that.

Defect Seeding is implanting the defect intentionally in the code to test the unit test case or specific condition or exception handling.

Different purposes of defect seeding:-

- To verify how the application is responding when the respective error occurs.

- Determine whether the build is properly tested or not.

- Determine the capability of testing experts in the QA team

Progressive testing, is quite functional. IN this you test the application with old test data. It is a first test which rolls down to regression test on retesting.
When an application with a hierarchy such as parent-child module is being tested, the related modules would need to be tested first.

Progressive testing also known as incremental testing is used to test modules one after the other.

When an application with a hierarchy such as parent-child module is being tested, the related modules would need to be tested first.

Progression testing focusses on new functionality and old functionalt and proving that it works as per the requirements as per hierarchy modules**.**
Whereas regression testing focuses on proving that existing functions of the application are not broken from the addition of new code.

- *Error* – mistake made by human during implementation of a software system.
- *Failure* – incorrect behaviour of a program.
- *Fault* – incorrect code that caused a failure.
- *Incident* – symptoms associated with a failure.
- *Bug* – an error or fault.
- *Fault Directed testing* – finding bugs through failure
- *Test Case* – A test described by test data, environment and expected result.
- *Test Suite* – a collection of test cases.
- *Test Plan* – document describing testing approach, test suites and test cases.
- *Test Strategy* – a way to identify test cases from a specification or implementation.