What is Manual Testing?

Manual testing is a testing process that is carried out manually in order to find defects without the usage of tools or automation scripting.

A test plan document is prepared that acts as a guide to the testing process in order to have the complete test coverage.

## What is Manual Testing?

Following are the testing techniques that are performed manually during the test life cycle:

* Acceptance Testing
* White Box Testing
* Black Box Testing
* Unit Testing
* System Testing
* Integration Testing

## What is Acceptance Testing?

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

* User acceptance Testing
* Business acceptance Testing
* Alpha Testing
* Beta Testing

## What is accessibility Testing?

Accessibility testing is a subset of usability testing where in the users under consideration are people with all abilities and disabilities. The significance of this testing is to verify both usability and accessibility.

Accessibility aims to cater people of different abilities such as:

* Visual Impairments
* Physical Impairment
* Hearing Impairment
* Cognitive Impairment
* Learning Impairment

A good web application should cater to all sets of people and NOT just limited to disabled people. These include:

1. Users with poor communications infrastructure
2. Older people and new users, who are often computer illiterate
3. Users using old system (NOT capable of running the latest software)
4. Users, who are using NON-Standard Equipment
5. Users, who are having restricted access

## What is Adhoc Testing?

When a software testing performed without proper planning and documentation, it is said to be Adhoc Testing. Such kind of tests are executed only once unless we uncover the defects.

Adhoc Tests are done after formal testing is performed on the application. Adhoc methods are the least formal type of testing as it is NOT a structured approach. Hence, defects found using this method are hard to replicate as there are no test cases aligned for those scenarios.

Testing is carried out with the knowledge of the tester about the application and the tester tests randomly without following the specifications/requirements. Hence the success of Adhoc testing depends upon the capability of the tester, who carries out the test. The tester has to find defects without any proper planning and documentation, solely based on tester's intuition.

## Forms of Adhoc Testing :

1. **Buddy Testing:**Two buddies, one from development team and one from test team mutually work on identifying defects in the same module. Buddy testing helps the testers develop better test cases while development team can also make design changes early. This kind of testing happens usually after completing the unit testing.
2. **Pair Testing:**Two testers are assigned the same modules and they share ideas and work on the same systems to find defects. One tester executes the tests while another tester records the notes on their findings.
3. **Monkey Testing:**Testing is performed randomly without any test cases in order to break the system.

## Various ways to make Adhoc Testing More Effective

1. **Preparation:** By getting the defect details of a similar application, the probability of finding defects in the application is more.
2. **Creating a Rough Idea:**By creating a rough idea in place the tester will have a focussed approach. It is NOT required to document a detailed plan as what to test and how to test.
3. **Divide and Rule:**By testing the application part by part, we will have a better focus and better understanding of the problems if any.
4. **Targeting Critical Functionalities:**A tester should target those areas that are NOT covered while designing test cases.
5. **Using Tools:**Defects can also be brought to the lime light by using profilers, debuggers and even task monitors. Hence being proficient in using these tools one can uncover several defects.
6. **Documenting the findings:** Though testing is performed randomly, it is better to document the tests if time permits and note down the deviations if any. If defects are found, corresponding test cases are created so that it helps the testers to retest the scenario.

What is Agile Testing?

A software testing practice that follows the principles of agile software development is called Agile Testing. Agile is an iterative development methodology, where requirements evolve through collaboration between the customer and self-organizing teams and agile aligns development with customer needs.

Advantages of Agile Testing

* Agile Testing Saves Time and Money
* Less Documentation
* Regular feedback from the end user
* Daily meetings can help to determine the issues well in advance

Principles of Agile Testing

* **Testing is NOT a Phase:** Agile team tests continuously and continuous testing is the only way to ensure continuous progress.
* **Testing Moves the project Forward:** When following conventional methods, testing is considered as quality gate but agile testing provide feedback on an ongoing basis and the product meets the business demands.
* **Everyone Tests:** In conventional SDLC, only test team tests while in agile including developers and BA's test the application.
* **Shortening Feedback Response Time:** In conventional SDLC, only during the acceptance testing, the Business team will get to know the product development, while in agile for each and every iteration, they are involved and continuous feedback shortens the feedback response time and cost involved in fixing is also less.
* **Clean Code:** Raised defects are fixed within the same iteration and thereby keeping the code clean.
* **Reduce Test Documentation:** Instead of very lengthy documentation, agile testers use reusable checklist, focus on the essence of the test rather than the incidental details.
* **Test Driven:** In conventional methods, testing is performed after implementation while in agile testing, testing is done while implementation.

Best Practices in Agile Testing

1. Automated Unit Tests

2. Test Driven Development

3. Automated Regression Tests

4. Exploratory Testing

Test-driven development (TDD) is a development technique where you must first write a test that fails before you write new functional code. TDD is being quickly adopted by agile software developers for development of application source code and is even being adopted by Agile DBAs for [database development](http://www.agiledata.org/essays/databaseTesting.html). TDD should be seen as complementary to [Agile Model Driven Development (AMDD)](http://www.agilemodeling.com/) approaches and the two can and should be used together. TDD does not replace traditional testing, instead it defines a proven way to ensure effective unit testing.