**Manual** **testing**: it is the process of testing the software to check whether the software is working properly or not based on the requirements by the client. The main aim of the tester is to find the defects in the software.

**Importance of testing:**

Testing is done achieve the following needs:

* Ensure software is bug free without defects
* Quality s/w to the end user
* Validates changes
* Reduces risks

**Common cause of the software bugs:**

the common cause of s/w bugs generally include the following such as:

* Specifications issues
* Design issues
* Environmental issues
* Integration issues
* Coding issues

**Software Testing:**

It is the process to detect and identify issues in the application/software. The objective of the testing is to deliver quality product to the customer.

**Software Quality:**

It can be defined as the quality of the software/application without the errors/bugs.

In simple terms it can be defined as the justification of all the requirement of the client.

**Product**: Product can be defined as the software/application developed for multiple customers based on the market requirement.

**Project**: Project can be defined as the software/application developed for the customer based on specific requirement.

**Why the s/w has bugs normally?**

The bugs can be formed due to several reasons such as:

* Miscommunication
* Software complexity
* Programming errors
* Lack of skilled testers

**SDLC(software development lifecycle):**

It the process/method used by the software professionals to design, develop and test the software/application. It basically consists of the stages such as requirement analysis, design , development , testing and maintenance.

**Waterfall** **model**: it is a type of s/w method or process in used to build a software.

**Advantages**:

* This is used for smaller projects
* Quality of the product will be good
* Initially no requirement of testers which reduces the cost of the project

**Disadvantages**:

* Additional requirements cannot be added later
* Testing will be added after coding only
* Defects are carried onto the next stages

The main purpose of the s/w testing is not to check or demonstrate the correctness of the project but to expose the hidden bugs or errors in the s/w (or) application

Testing gives the number of errors/bugs present in the s/w which in turn gives the quality of the product. More number of bugs/defects indicate bad quality and bad development process

The main cause of the s/w bug is the “**specification**”, i.e. the requirement varies constantly

The next largest source for the bug is the “**design**”, if you cannot design the s/w correctly then it is prone to maximum bugs

**Cost of the bugs:**

The cost of fixing bugs is logarithmic i.e. the cost of fixing the bug increases as time is increased. Early detection of the bugs reduces the cost while the bugs found later or detected at post release can be expensive

“**The main goal of the software tester is to find the bugs as soon as possible and fix it as early as possible”**

**Software testing:**

The process of testing the s/w to find an error in it.

**Software Testing types:**

The testing is of two types namely :

1. Manual testing
2. Automation testing

Manual testing: it is the process in which the testing is done without the use of an automated tools or script.

**Script:** the script is a set of instructions written in a programming language or scripting language that tells the automation tool what actions to perform while testing

**Automation** **testing**: it is the process of testing in which the tester writes the script and uses another s/w to test the product. It is the process of automation of the manual process

The automation testing use tools and scripts for automation such as it automates the repetitive tasks , it ensures faster execution

**Example for manual and automation testing:**

**Login functionality of a website:** using manual testing the details such as username and password needs to be entered manually and clicking the login page to check the functionality, **but** by using the automation testing a script is written to check the functionality.

**Selenium**: selenium is both a tool and a package(contains modules) that is used to test the website functionality, compatibility and performance

**Software testing classification:** software testing is broadly classified into two categories of how and when testing is done.

1. **Based on execution:**

**Static testing:**  this testing is done before the execution of the code; it involves reviewing the documents and articrafts to find potential defects early. The static techniques include reviews (informal, walkthrough, inspection)

**Dynamic testing:** it is done by executing the software to check the behavior during runtime

1. **Based on testers knowledge:**
2. **Whitebox testing:** in this testing the tester has full knowledge about the internal architecture, code and algorithms (testing some small sort of code such as if-else statements)
3. **Blackbox testing:** in this testing the tester has zero knowledge about the internal architecture, code (testing the login page without having the knowledge about the backend)
4. **Gray box testing:** in this testing the tester has limited knowledge about the s/w (API testing)

**3.Based on Levels of testing:**

**Unit testing:** in this testing the testing is done on individual component, it focuses on smaller unit of the s/w design. This is usually done by developers by providing a sample input and checking the output.

**Integration testing:** in this testing process the testing is performed by joining the modules into the group. The main purpose of the integration testing is to check whether the modules are communicating with each other properly. This testing is also done by the developer.

**System testing:** in system testing the testing is done on whole the software application. It is usually done by the tester.

**User acceptance testing:** a level of software testing in which testing is done for user acceptance.

**4.Based on objective:**

**Functional testing:** it is the type of testing in which the system is tested against the functional requirements and specifications

**Non functional testing :** it is the type of testing in which the system is tested against the non-functional requirements and specifications.

The functional testing is further divided into :

* Unit testing
* Integration testing
* System testing
* End to end testing
* Acceptance testing

The non functional testing is divided into :

* Performance testing (load, stress, scalability, stability)
* Usability testing
* Compatibility testing

**Regression testing:** it is a type of testing that ensures the code changes doesn’t negatively affect the existing features of the s/w

**Sanity testing:** Sanitytestingensures that the specific functionality of the s/w works properly

**Smoke testing:** this ensures that the basic functionalities of the s/w works properly (search bar, landing homepage)

The above regression, sanity, smoke testing also comes under the types of testing “**based on objective”**

**Test case design techniques or black box design techniques:**

The test case design techniques are classified as:

* **Error guessing:** in this design the test engineer will think of all the negative possibilities of the system

**Ex:** assume that the test field requirement is to enter only the positive integers, then the test engineer will check for all the negative possibilities such as by entering alphanumeric, negative numbers, alphabets as the input

* **Equivalence** **partitioning**: this design is also called as equivalence **class** partitioning. In this design the input is divided into sub classes and from each class a random input is selected and verified for each class

**Ex**: assume you have a requirement to enter the values only between 1 to 500. In this case instead of entering the inputs manually from 1-500 since this is a time consuming process the 500 inputs and divided into groups called classes such as (-100 to 0 , 1-100 , 101-200….so on) and from this classes one input is selected and if that test case pass the whole class is present and there is no need to check the other value from the class. In this way the ECP saves the tester time

* **Boundary value analysis:** as the name suggests it the testing at the boundaries between the partitions.

**Ex :** assume you have two numbers namely **a** and **b ,** in this case the test cases for **a** would be (**a-1 , a , a+1**) and for **b** would be (**b-1 , b , b+1**)

* **Decision table technique:** it is technique in which the test cases are designed based on condition, combination and rule criteria

**Ex:** assume the following:

**New\_customer = 15%**

**Repeating\_customer = 10%**

**Coupon\_code = 20%**

**Therefore,** the no. of test cases = no. of rules = 2^ no. of conditions

**For above, = 2^3 = 8**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Conditions/Rules | Rule1 | Rule2 | Rule3 | Rule4 | Rule5 | Rule6 | Rule7 | Rule8 |
| **15%** | T | T | T | T | F | F | F | F |
| **10%** | T | T | F | F | T | T | F | F |
| **20%** | T | F | T | F | T | F | T | F |
| **Result** | x | x | 35% | 15% | 30% | 10% | x | x |

* **State transition technique:** it is technique in which the systems behavior depends on the current state. It tests the transitions between the different stages triggered by the various inputs. As the name says transitions from one stage to another

**Ex:** food delivery app and if user enters the login credentials wrong for more than three times then the user should be blocked and doesn’t allow him to order the food.