**Phase 1- Software Testing Concepts**  
What is Software?  
Types of Software  
What is Testing?  
What is Software Testing?  
Why Software Testing is Important?  
What is Software Quality?  
Why the Software has bug?  
Error, Bug/Defect, Fault & Failure  
What is Manual Testing?  
Why we need manual testing?  
How to perform Manual Testing?  
Types of Manual Testing  
advantage & Disadvantage of Manual  
Testing  
Goal of Manual Testing  
7 Principal of Software Testing   
Software Testing Types  
Product & Project  
QA/Testing Activities  
  
**Phase 2 - Software Testing Project**  
Project Introduction  
Understanding Requirement (BRS, Mockup,  
Wireframe)  
Understanding on Test Scenario & Test Case  
Test Execution  
-------------------------------------------------------------------------------------------  
  
What is Software?  
  
Software is a collection of computer program or set of programs (sequence of instructions) that allows the users to perform some specified task.  
  
software programmers write the software program in various human-readable languages such as Java, Python, C#, etc. and later use the source code.  
  
A program is a sequence of instructions or a piece of code written to solve a particular problem.  
  
Types of software:  
There are two types of software −  
  
1. System Software  
2. Application Software  
  
System Software:  
  
System software is a computer program that helps the user to run computer hardware or software and manages the interaction between them. Essentially, it is software that constantly runs in the computer background, maintaining the computer hardware and computer's basic functionalities, including the operating system, utility software, and interface. In simple terms, we can say that the system acts as a middle man that checks and facilitates the operations flowing between the user and the computer hardware.  
  
Example: Operating System (Microsoft Windows), Device Drivers (Motherboard Drivers), Utility(Windows File Explorer)  
  
Application Software:  
  
Application programs or software applications are end-user computer programs developed primarily to provide specific functionality to the user.  
  
The application software can either be designed for a general-purpose or specially coded as per the requirements of business cooperation.  
  
Example: Web Application, Mobile Application, Desktop Application  
  
What is Testing?  
  
Testing is a group of techniques to determine the correctness of the application under the predefined script but, testing cannot find all the defect of application. The main intent of testing is to detect failures of the application so that failures can be discovered and corrected. It does not demonstrate that a product functions properly under all conditions but only that it is not working in some specific conditions.  
  
What is Software Testing?  
  
Software Testing is an activity to detect and identify the defects in the software.  
  
The objective of testing is to release quality release to the client  
  
Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free.  
  
The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.  
  
Why Software Testing is Important?  
  
Software Testing is Important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.  
  
Software Quality (based on some parameter):  
  
Bug free  
Delivered on time  
Within budget  
Meets requirement and /or expectation  
Maintainable  
  
Error, Bug/Defect, Fault & Failure:  
  
What is a bug?  
In software testing, a bug is the informal name of defects, which means that software or application is not working as per the requirement. When we have some coding error, it leads a program to its breakdown, which is known as a bug. The test engineers use the terminology Bug.  
  
If testers find any mismatch in the application/system in testing phase then they call it as Bug.  
  
If that defect is accepted by the developer then it will be called a bug.  
  
  
Note: there is a contradiction in the usage of Bug and Defect. People widely say the bug is an informal name for the defect.  
  
What is Defect?  
When the application is not working as per the requirement is knows as defects. It is specified as the aberration from the actual and expected result of the application or software.  
  
In other words, we can say that the bug announced by the programmer and inside the code is called a Defect.  
  
error found by tester called defect  
  
If a developer finds an issue and corrects it by himself in the development phase then it’s called a defect.  
  
What is Error?  
  
We can’t compile or run a program due to coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an .  
  
A mistake in coding is called error  
  
The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly.   
The developers use the term error.  
  
What is Fault?  
  
The fault may occur in software because it has not added the code for fault tolerance, making an application act up.  
We can’t compile or run a program due to coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an .  
  
**What is Failure?**  
  
Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as   
  
 if an end-user detects an issue in the product, then that particular issue is called a failure.  
  
For example, in a bank application if the Amount Transfer module is not working for end-users when the end-user tries to transfer money, submit button is not working. Hence, this is a failure.  
  
Why the Software has bug?  
  
There are many reasons for the occurrence of Software Bugs. The most common reason is human mistakes in software design and coding.  
  
**Top some Reasons for Software Bugs:**  
  
Miscommunication or No Communication  
  
Software Complexity  
  
Programming Errors  
  
Changing Requirements  
  
Poorly Documented  
  
Lack of Skilled Testers  
  
Bad coding  
  
**What is Manual Testing?**  
Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application. Manual software testing is the most primitive technique of all testing types and it helps to find critical bugs in the software application.  
  
Any new application must be manually tested before its testing can be automated.  
  
Manual Testing concepts does not require knowledge of any testing tool.   
  
**Why we need manual testing?**  
Whenever an application comes into the market, and it is unstable or having a bug or issues or creating a problem while end-users are using it.  
If we don't want to face these kinds of problems, we need to perform one round of testing to make the application bug free and stable and deliver a quality product to the client, because if the application is bug free, the end-user will use the application more conveniently.  
If the test engineer does manual testing, he/she can test the application as an end-user perspective and get more familiar with the product, which helps them to write the correct test cases of the application and give the quick feedback of the application.  
  
**How to perform Manual Testing?**  
Analyze requirements from the software requirement specification document   
Create a clear test plan  
Write test cases that cover all the requirements defined in the document  
Get test cases reviewed by the QA lead  
Execute test cases and detect any bugs  
Report bugs, if any, and once fixed, run the failed tests again to re-verify the fixes  
  
**Types of Manual Testing**

* Black Box Testing
* System Testing
* Integration Testing
* Acceptance Testing

**Advantages of Manual Testing**  
  
No Environment Limitations  
Programming Knowledge is not required  
Recommendable for Dynamically changing GUI designs  
Recommendable for Usability Testing  
Manual testing allows for human observation, which may be more useful to find potential defect  
More reliable than automation  
Easy to learn for new people who has just entered into testing  
Exploratory testing can be done  
It is not possible to cover all negative scenarios / random scenarios so we prefer manual testing  
Best for small scale application  
  
**Disadvantages of Manual Testing**  
Manual Testing requires more time or more resources, sometimes both Time and Resources  
Executing the same tests, again and again, is time taking process as well as Tedious  
For every release you must rerun the same set of tests which can be tiresome  
It is very time-consuming  
It does not cover all aspects of testing  
  
**Goal of Manual Testing**  
  
The key concept of manual testing is to ensure that the application is error free and it is working in conformance to the specified functional requirements.  
Test Suites or cases, are designed during the testing phase and should have 100% test coverage.  
It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects.  
Basically, this testing checks the quality of the system and delivers bug-free product to the customer  
  
**7 Principles of Software Testing**  
The seven basic Software Testing Principles that every Software tester and QA professional should know.  
  
7 Principles of Software Testing:  
1. Testing shows presence of defects  
  
There are several domains available in the market like Banking, Insurance, Medical, Travel, Advertisement etc and each domain has a number of applications. Also for each domain, their applications have different requirements, functions, different testing purpose, risk, techniques etc.  
Different domains are tested differently, thus testing is purely based on the context of the domain or application.  
For Example, testing a banking application is different than testing any e-commerce or advertising application. The risk associated with each type of application is different, thus it is not effective to use the same method, technique, and testing type to test all types of application.  
2. Exhaustive testing is not possible  
  
It is not possible to test all the functionalities with all valid and invalid combinations of input data during actual testing. Instead of this approach, testing of a few combinations is considered based on priority using different techniques.  
Exhaustive testing will take unlimited efforts and most of those efforts are ineffective. Also, the project timelines will not allow testing of so many number of combinations. Hence it is recommended to test input data using different methods like Equivalence Partitioning and Boundary Value Analysis.  
3. Early testing  
  
Early Testing – Testing should start as early as possible in the Software Development Life Cycle. So that any defects in the requirements or design phase are captured in early stages. It is much cheaper to fix a Defect in the early stages of testing. But how early one should start testing? It is recommended that you start finding the bug the moment the requirements are defined  
4. Defect clustering  
  
During testing, it may happen that most of the defects found are related to a small number of modules. There might be multiple reasons for this like the modules may be complex, coding related to such modules may be complicated etc.  
This is the Pareto Principle of software testing where 80% of the problems are found in 20% of the modules.   
By experience, we can identify such risky modules.  
5. Pesticide paradox  
Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs.  
6. Testing is context dependent  
The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, The testing of the e-commerce site is different from the testing of the Android application.  
7. Absence of errors fallacy  
  
If the software is tested fully and if no defects are found before release, then we can say that the software is 99% defect free. But what if this software is tested against wrong requirements? In such cases, even finding defects and fixing them on time would not help as testing is performed on wrong requirements which are not as per needs of the end user.  
  
For Example, suppose the application is related to an e-commerce site and the requirements against “Shopping Cart or Shopping Basket” functionality which is wrongly interpreted and tested. Here, even finding more defects does not help to move the application into the next phase or in the production environment.  
  
**Software Testing Types:**  
White Box Testing & Black Box Testing  
Positive Testing   
Negative Testing  
Exploratory Testing  
Ad hoc testing  
Re-Testing  
Regression Testing   
Smoke Testing  
Sanity Testing  
Smoke Testing   
System Testing  
Integration Testing   
Usability Testing  
  
**Product & Project:**  
if Software application is developed for specific customer based on the requirements then it is called Project.  
  
Example: banking Application  
  
if Software application is developed for multiple customer based on the market requirements then it is called Product.  
  
Example: MS Excel, Word, Google product (Google Map)  
  
**QA/Testing Activities:**  
  
Understanding the requirement functional specifications of the applications  
Identifying required test scenarios'  
Designing test cases to validate application  
Setting up test environment  
Execute test case to validate requirement  
Log Test Result (How many test cases pass or fail)  
Defect reporting and tracking  
Retesting fixed defect which is find out during testing  
Perform various types of testing in the application  
Report to respective line manager regarding the assigned task status  
Participate on regular team meeting (scrum meeting)  
Creating automation script  
provides recommendation on whether or not the application is ready for production