DUY TAN UNIVESITY

SOFTWARE TESTING

ASSIGNMENT

Teacher : Dr. Anand Nayyar

Student : Tuan, Pham Minh

Class : K22MCS

Da Nang, 02/2021

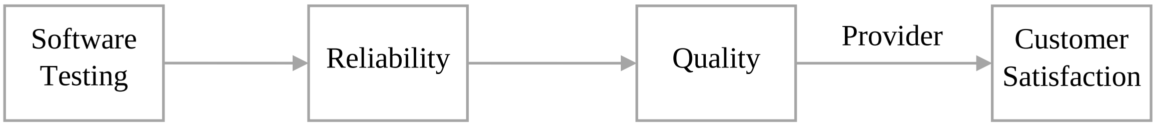
QUESTION 1: Explain testing with perspective towards improvement of Customer Satisfaction in Industry? Highlight the role of Software Tester? Elaborate: Alpha, Beta, White-box, Gray Box and Black Box Testing? Write 5 Differences between White Box and Black Box Testing?

- **Explain testing with perspective towards improvement of Customer Satisfaction in Industry?**

Customer satisfaction is one long-term goals of software testing. From the user's perspective, the prime goal of software testing is customer satisfaction only. If we want the client and customer to be satisfied with the software product, then testing should be complete and thorough.

Testing should be complete in the sense that it must satisfy the user for all the specified requirements mentioned in the user manual, as well as for the unspecified requirements, which are otherwise understood.

A complete testing process achieves reliability, reliability enhances the quality, and quality in turn, increases the customer satisfaction.



- **Highlight the role of Software Tester**

The delivery of an optimal quality software product that has unique and innovative features has always been the priority of the software industry worldwide. However, without evaluating software components under various expected and unexpected conditions, the team cannot guarantee these aspects. Therefore, testing is performed to test every software component large and small.

To understand Software testing importance, let's look into the below points:

The software test is essential. So, do not start over from scratch again: Sometimes, we test a fully developed software product against the user requirement and find that some basic functionality was missing. It may happen because of a mistake in the requirement gathering or the coding phase. Then to fix such types of errors, we may have to start the development again from scratch. Fixing such kinds of mistakes becomes very tedious, time-consuming, and expensive. Therefore, it is always desirable to test the software in its development phase.

Evaluating the ease of use of the software: Ease of use is a simple concept; it specifies how easily the intended users can use the final product. The software testing ensures the construction of the software product in a way that meets the user’s expectations regarding compliance with the requirements in a comfortable, satisfactory, and simplistic manner.

Verification of all aspects of the software: You can verify all the aspects of the software in software testing, such as checking the basic functionalities as well as testing a system for unexpected conditions. Unexpected conditions can be from an incorrect data type or due to a piracy attack. Therefore, testing makes sure that the system can handle these situations very well. Thus, if we find an error in advance, we have the option to correct them. It can prevent complaints once the software or application has reached customers.

Software tests help accelerate development: Software tests help developers find errors and scenarios to reproduce the error, which in turn helps them to fix it quickly. Besides, software testers can work in parallel with the development team, thus understanding the design, risk areas, etc. in detail. This knowledge exchange between testers and developers accelerates the entire development process.

Software testing is a crucial component of software product development because it improves consistency and performance. The main benefit of testing is the identification and subsequent removal of the errors. However, testing also helps developers and testers to compare actual and expected results in order to improve quality. If the software production happens without testing it, it could be useless or sometimes dangerous for customers. So, a tester should wear a unique hat that protects the reliability of the software and makes it safe to use in real-life scenarios.

❖ Elaborate: Alpha, Beta, White-box, Gray Box and Black Box Testing?

Alpha Testing

Alpha Testing is a type of acceptance testing. This type of testing does not involve functional testing on the application. Instead, it is a user testing on the application in order to understand the user behavior and experience on the application.

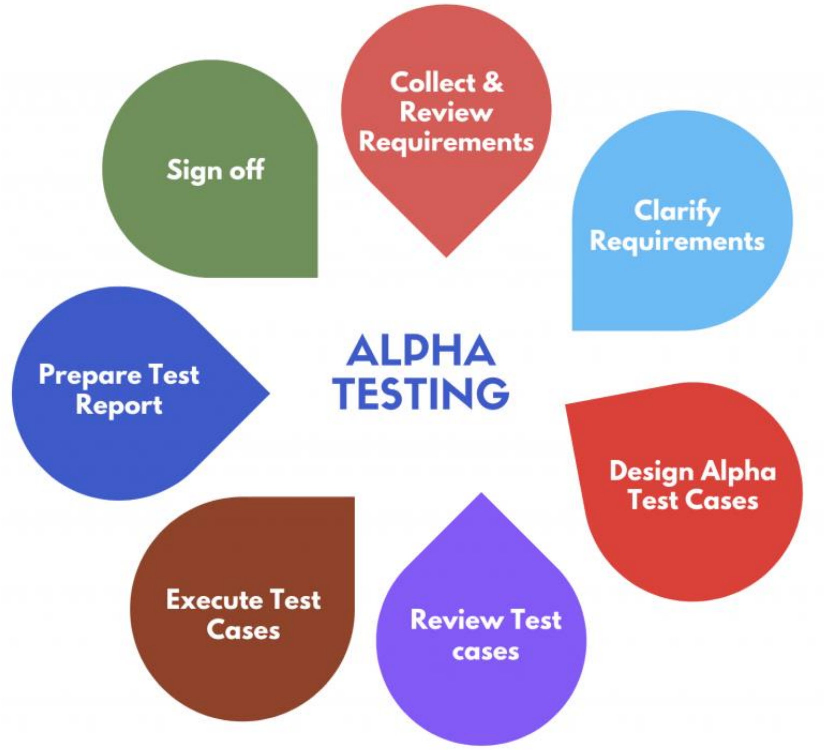
Normally this test is performed by test engineers, employees and sometimes friends/family members with the aim of trying to emulate around 80% of the customers. While these users test and give their feedback, the development team observes the behavior to check for design issues in the application.

Alpha testing is mainly conducted to unveil bugs that might arise due to abrupt errors created by the users, validate the quality state of the software in minimal time and finally propound a build that procures the specifications required. Once this test is executed properly, the software is ready for the next stage, i.e., the beta test.

Alpha testing has two phases. The first phase consists of testing by the developers. The software used is either hardware-assisted debuggers or debugger software. The basic motive is to detect bugs quickly. The second phase of testing is done by the quality assurance team, which ensures that the build works perfectly on the environment similar to user end.

The procedure to conduct procedure testing is as follows:

- Collect and review the functional requirements with the product documentation.
- After reviewing, clarify the questions regarding the functional requirements.
- Design Alpha test cases.
- Review the Test cases for coverage, traceability, and the essential features of the product.
- Once the product is ready for Alpha testing, execute the designed Alpha test cases.
- Prepare the Alpha test report.
- Alpha test formal sign off.



Beta Testing

Beta Testing is also known as field testing. It takes place at customer's site. It sends the system/software to users who install it and use it under real-world working conditions.

This is one of the Customer Validation methodologies to evaluate the level of customer satisfaction with the product by letting it to be validated by the end users, who actually use it, for over a period of time.

A beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. (Beta is the second letter of the Greek alphabet.) Originally, the term alpha testing meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. Beta testing can be considered "pre-release testing.

The goal of beta testing is to place your application in the hands of real users outside of your own engineering team to discover any flaws or issues from the user's perspective that you would not want to have in your final, released version of the application.

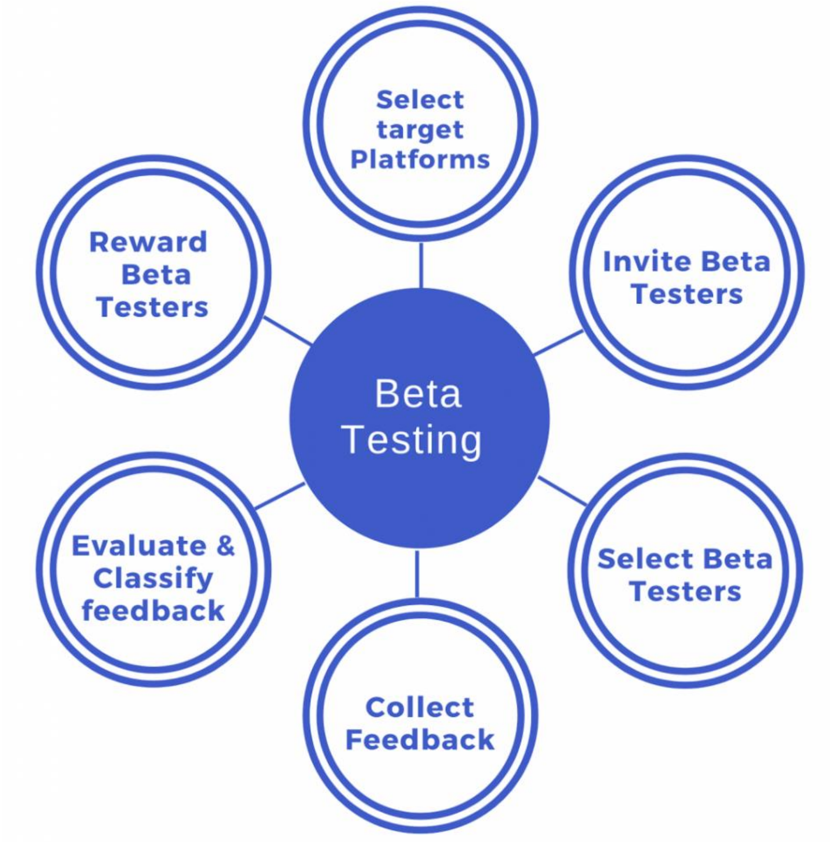
Example: Microsoft and many other organizations release beta versions of their products to be tested by users.

The procedure to conduct Beta testing is as follows:

- Select the target platforms
- Invite Beta testers
- Select Beta testers within the various environments
- Collect feedback
- Evaluate and classify the outcomes of Beta testing
- Reward Beta testers

Advantages of beta testing:

- You have the opportunity to get your application into the hands of users prior to releasing it to the general public.
- Users can install, test your application, and send feedback to you during this beta testing period.
- Your beta testers can discover issues with your application that you may have not noticed, such as confusing application flow, and even crashes.
- Using the feedback you get from these users, you can fix problems before it is released to the general public.
- The more issues you fix that solve real user problems, the higher the quality of your application when you release it to the general public.
- Having a higher-quality application when you release to the general public will increase customer satisfaction.
- These users, who are early adopters of your application, will generate excitement about your application.



White Box Testing

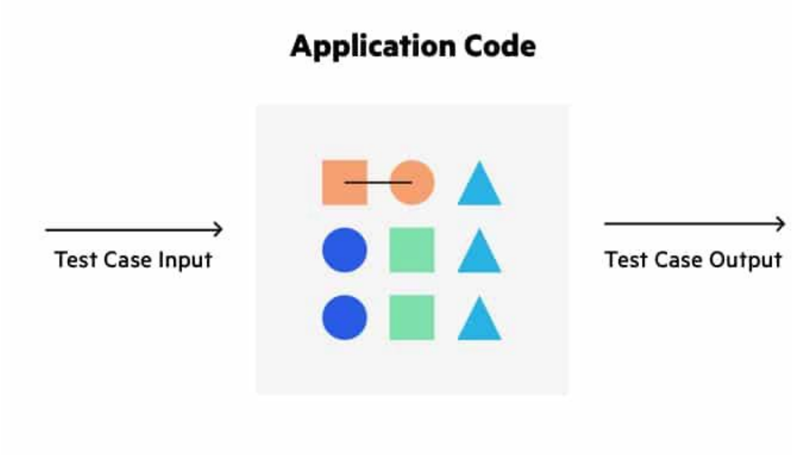
White box testing is an approach that allows testers to inspect and verify the inner workings of a software system - its code, infrastructure, and integrations with external systems. In this method, testing is based on coverage of code statements, branches, paths or conditions. White-Box testing is considered as low-level testing. It is also called glass box, transparent box, clear box or code base testing. The white-box Testing method assumes that the path of the logic in a unit or program is known.

White Box Testing is designed to do test all possible scenarios that a system can encounter. By testing at the source code-level, a tester will be able to run through every single permutation and combination that the program can theoretically spew out.

When testing happens at such granular level, this then brings any possible defects out in the open. And your team will have an opportunity to evaluate whether some or all of them need to be fixed.

Working process of white box testing:

- Input: Requirements, Functional specifications, design documents, source code.
- Processing: Performing risk analysis for guiding through the entire process.
- Proper test planning: Designing test cases so as to cover entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
- Output: Preparing final report of the entire testing process.



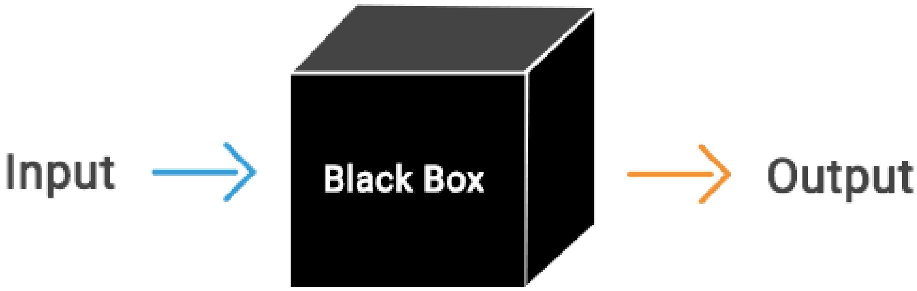
Black Box Testing

In Black-box testing, a tester doesn't have any information about the internal working of the software system. Black box testing is a high level of testing that focuses on the behavior of the software. It involves testing from an external or end-user perspective. Black box testing can be applied to virtually every level of software testing: unit, integration, system, and acceptance.

Following are some techniques that can be used for designing black box tests:

- Equivalence Partitioning: It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.
- Boundary Value Analysis: It is a software test design technique that involves the determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.

- Cause-Effect Graphing: It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.



Gray Box Testing

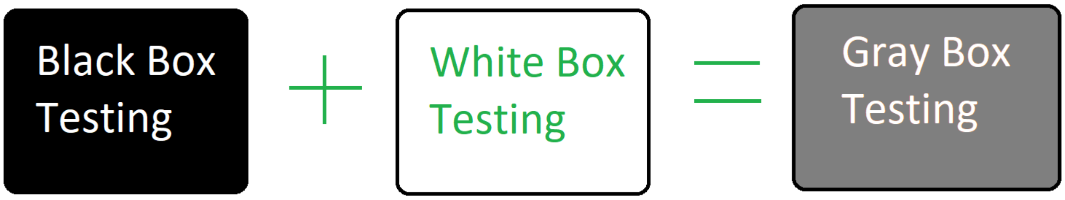
Grey Box Testing or Gray box testing is a software testing technique to test a software product or application with partial knowledge of internal structure of the application. The purpose of grey box testing is to search and identify the defects due to improper code structure or improper use of applications.

In this process, context-specific errors that are related to web systems are commonly identified. It increases the testing coverage by concentrating on all of the layers of any complex system.

Gray Box Testing is a software testing method, which is a combination of both White Box Testing and Black Box Testing method.

Gray Box Testing is performed for the following reason:

- It provides combined benefits of both black box testing and white box testing both
- It combines the input of developers as well as testers and improves overall product quality
- It reduces the overhead of long process of testing functional and non-functional types
- It gives enough free time for a developer to fix defects
- Testing is done from the user point of view rather than a designer point of view



Example:

An example of Gray Box Testing would be when the codes for two units/modules are studied (White Box Testing method) for designing test cases and actual tests are conducted using the exposed interfaces (Black Box Testing method).

❖ **5 differences between Black Box Testing vs White Box Testing:**

Black Box Testing	**White Box Testing**
No knowledge of implementation is needed	Knowledge of implementation is required
It is the behavior testing of the software	It is the logic testing of the software.
Mainly applicable to higher levels of testing: Acceptance Testing & System Testing	Mainly applicable to lower levels of testing: Unit Testing & Integration Testing
This testing can be initiated on the basis of requirement specifications document	This type of testing of software is started after detail design document
It is least time consuming.	It is most time consuming.

QUESTION 2: What is Web Testing with perspective to security and confidentiality of Data by maintaining Integrity? What are the various Web Testing Techniques and How you differentiate Web Testing with Penetration Testing?

❖ What is Web Testing with perspective to security and confidentiality of Data by maintaining Integrity?

Data integrity is a fundamental component of information security. In its broadest use, "data integrity" refers to the accuracy and consistency of data stored in a database, data warehouse, data mart or other construct. The term – Data Integrity **-** can be used to describe a state, a process or a function – and is often used as a proxy for "data quality". Data with "integrity" is said to have a complete or whole structure. Data values are standardized according to a data model and/or data type. All characteristics of the data must be correct – including business rules, relations, dates, definitions and lineage – for data to be complete. Data integrity is imposed within a database when it is designed and is authenticated through the ongoing use of error checking and validation routines. As a simple example, to maintain data integrity numeric columns/cells should not accept alphabetic data.

❖ What are the various Web Testing Techniques?

Web application testing, a software testing technique exclusively adopted to test the applications that are hosted on web in which the application interfaces and other functionalities are tested.

Functional Testing: The below are some of the checks that are performed but not limited to the below list:

- Verify there is no dead page or invalid redirects.
- First check all the validations on each field.
- Wrong inputs to perform negative testing.
- Verify the workflow of the system.
- Verify the data integrity.

Usability Testing: To verify how the application is easy to use with:

- Test the navigation and controls.
- Content checking.
- Check for user intuition.

Interface testing: Performed to verify the interface and the dataflow from one system to other.

Compatibility testing: Compatibility testing is performed based on the context of the application.

- Browser compatibility
- Operating system compatibility
- Compatible to various devices like notebook, mobile, etc.

Performance testing: Performed to verify the server response time and throughput under various load conditions.

- Load testing
- Stress testing
- Soak testing
- Spike testing

Security testing: Performed to verify if the application is secured on web as data theft and unauthorized access are more common issues and below are some of the techniques to verify the security level of the system.

- Injection
- Broken Authentication and Session Management
- Cross-Site Scripting (XSS)
- Insecure Direct Object References
- Security Misconfiguration
- Sensitive Data Exposure
- Missing Function Level Access Control
- Cross-Site Request Forgery (CSRF)
- Using Components with Known Vulnerabilities

- Unvalidated Redirects and Forwards

❖ How you differentiate Web Testing with Penetration Testing?

Web Testing is: the practice of thoroughly checking over a website or web application either before it has launched or once it is live on the web and accessible to the public. The general aim of web testing is to find problems, also known as bugs, issues or defects, that might have a negative effect on the website or application.

Web Testing checklist below:

- Functionality Testing
- Usability Testing
- Interface Testing
- Compatibility Testing
- Performance Testing
- Security Testing

Penetration Testing is: an authorized simulated attack performed on a computer system to evaluate its security. Penetration testers use the same tools, techniques, and processes as attackers to find and demonstrate the business impacts of weaknesses in your systems.

Penetration Testing checklist below:

- Black hat attack methodologies (e.g., remote access attacks, SQL injection)
- Internal and external testing (i.e., perspective of someone within the network, perspective of hacker over Internet)
- Web front-end technologies (e.g.,Javascript, HTML)
- Web application programming languages (e.g., Python, PHP)
- Web APIs (e.g., restful, SOAP)
- Network technologies (e.g, firewalls, switches, IDS)
- Networking protocols (e.g., TCP/UDP, SSL)
- Operating systems (e.g., Linux, Windows)
- Scripting languages (e.g., Python, Perl)

- Testing tools (e.g., Nessus, Metasploit)

QUESTION 3: Design some Test Cases for following: Smart Phone Battery, Web Shopping Cart, Smartphone Sensors - Any 4 Sensors

❖ Testcase for Smart Phone Battery

- Plug in the charger, does the phone show the charge status and battery percentage in the status bar?
- When charging the phone, is the phone hotter than the recommended temperature?
- When the phone plugs the charger according to the recommended time, please check if the battery capacity is full?
- When the phone battery is full, use the phone to browse the web and read news ... in the recommended time, will the phone battery drain faster than the manufacturer recommends?
- When the phone battery is full, use the phone to play games in the recommended time, will the phone drain the battery faster than the manufacturer recommends?
- When the phone shows 0 percent battery, will the phone turn off after a few seconds?

❖ Testcase for Web Shopping Cart

- Add one item to the cart and verify.
- Increase the quantity of the item from the cart and verify.
- Add the same item multiple times and verify.
- Add multiple items of different types and verify.
- Remove some items from the cart and verify.
- Remove all items from the cart and then verify.
- Click on an item in the cart and verify that the user is redirected to the product detail page.
- Check that the price of the cart is discounted when we apply a valid coupon.
- Check that the price of the cart is not discounted when we apply an invalid coupon.

- Add item(s) to the cart, close the browser and reopen the same site.
- Add item(s) to the cart, close the browser and reopen the same site.
- Verify the product QTY field when the product is out of stock.
- Verify that the user is able to add a text note for all products.
- Verify that the user is able to add any or all products to his wishlist by clicking on the wishlist link.
- Verify that the user is able to mark his order as a gift.
- Verify that the user is able to add any a gift message.
- Verify that the user is redirected to the checkout page after clicking on the checkout button.
- Verify the cart total when the exclusive tax is enabled from the admin end.
- Verify the cart total when the inclusive tax is enabled from the admin end.

❖ Testcase for Smartphone Proximity Sensor

- When you hold the phone to your ear to call or answer, does the phone screen turn off?
- When you lift the phone out of your ear during a call or answer, does the phone screen turn on?
- When the phone is locked, you hold the phone to your ear and then bring it down, does the phone screen turn on?

QUESTION 4: Highlight Software Quality Assurance with perspective to IT and Telecom Industry? What is the procedure associated with it? And how you assure the quality of the software design and development?

❖ Highlight Software Quality Assurance with perspective to IT and Telecom Industry?

Software quality assurance (SQA) is a process which assures that all software engineering processes, methods, activities and work items are monitored and comply against the defined standards. These defined standards could be one or a combination of any like ISO 9000, CMMI model, ISO15504, etc.

SQA incorporates all software development processes starting from defining requirements to coding until release. Its prime goal is to ensure quality.

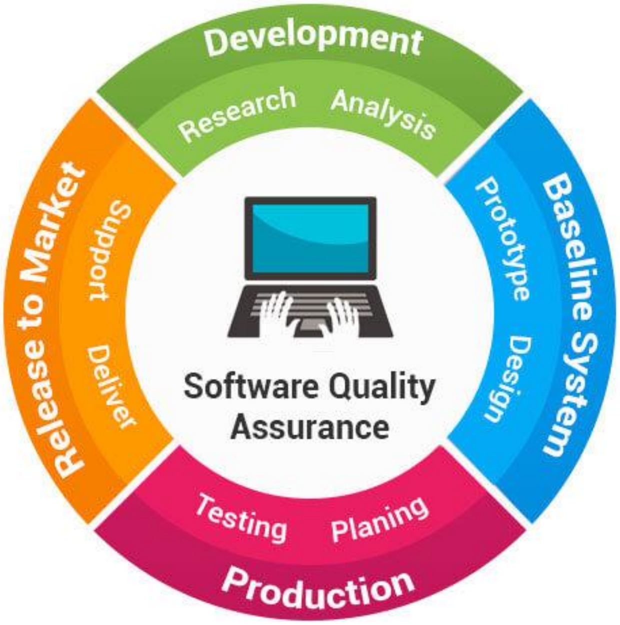
QA ensures the implementation of standard requirements and procedures, the planned course of action by the QA team guarantees product quality and success.

QA helps the entire software team in terms of reporting paperwork, documentation, and associated data.

QA ensures that all essential procedures are being followed throughout the entire development cycle.

QA team performs activities such as testing, process monitoring, audits, as well as product evaluation to improve productivity.

QA gives a transparent outlook of collaboration and management for DevOps teams.



Software Quality Assurance Plan

❖ **What is the procedure associated with it?**

- Review of requirements
- Test planning/writing test cases

- Unit testing
- Integration testing
- System testing
- Performance testing
- Security testing
- Cross-browser testing / cross-platform testing
- Updating test cases
- Regression testing

❖ And how you assure the quality of the software design and development?

Software quality is defined as a field of study and practice that describes the desirable attributes of software products. There are two main approaches to software quality: defect management and quality attributes.

Software Quality Defect Management Approach: A software defect can be regarded as any failure to address end-user requirements. Common defects include missed or misunderstood requirements and errors in design, functional logic, data relationships, process timing, validity checking, and coding errors.

The software defect management approach is based on counting and managing defects. Defects are commonly categorized by severity, and the numbers in each category are used for planning. More mature software development organizations use tools, such as defect leakage matrices (for counting the numbers of defects that pass through development phases prior to detection) and control charts, to measure and improve development process capability.

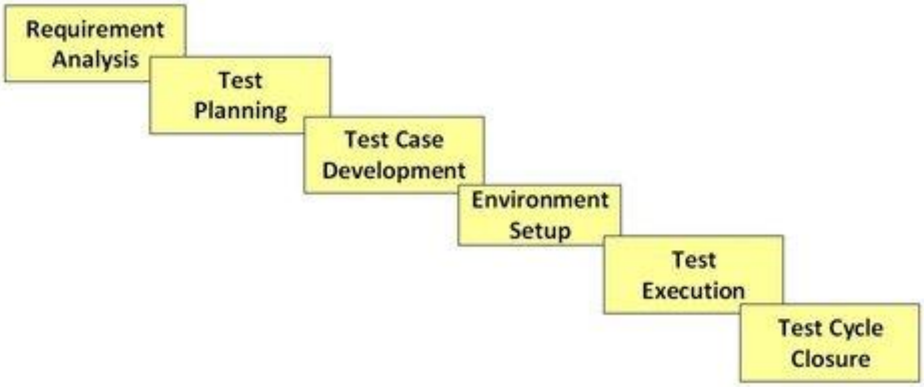
Software Quality Attributes Approach: This approach to software quality is best exemplified by fixed quality models, such as ISO/IEC 25010:2011. This standard describes a hierarchy of eight quality characteristics, each composed of sub-characteristics:

- Functional suitability
- Reliability

- Operability
- Performance efficiency
- Security
- Compatibility
- Maintainability
- Transferability

QUESTION 5: Explain Software Testing Life Cycle with a Case Study?

There are following six major phases in every Software Testing Life Cycle:



- Requirement Analysis: Requirement Analysis in which test team studies the requirements from a testing point of view to identify testable requirements and the QA team may interact with various stakeholders to understand requirements in detail. Requirements could be either functional or non-functional. Automation feasibility for the testing project is also done in this stage.
- Test Planning: Test Planning in Software Testing Life Cycle is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project. Moreover, the resources, test environment, test limitations and the testing schedule are also determined. The Test Plan gets prepared and finalized in the same phase.

- Test case development: The Test Case Development Phase involves the creation, verification and rework of test cases & test scripts after the test plan is ready. Initially, the Test data is identified then created and reviewed and then reworked based on the preconditions. Then the QA team starts the development process of test cases for individual units.
- Test Environment setup: Test Environment Setup decides the software and hardware conditions under which a work product is tested. It is one of the critical aspects of the testing process and can be done in parallel with the Test Case Development Phase. Test team may not be involved in this activity if the development team provides the test environment. The test team is required to do a readiness check (smoke testing) of the given environment.
- Test Execution: Test Execution Phase is carried out by the testers in which testing of the software build is done based on test plans and test cases prepared. The process consists of test script execution, test script maintenance and bug reporting. If bugs are reported then it is reverted back to development team for correction and retesting will be performed.
- Test Cycle closure: Test Cycle Closure phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices and test results. Testing team members meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from current test cycle. The idea is to remove process bottlenecks for future test cycles.