**Boot Camp 2020**

**Teacher : AnkurJain**

**20221MD**

**Emre Baybas**

* **What Is the Difference Between Sdlc And Stlc?**

The main object of SDLC is to complete successful development of the software that meets the requirements.

STLC is the integral part of the SDLC that deals with the test related phases of the software development.

* **What Is Stlc?**

STLC is types of activities that are perform to ensure the quality of the product.

It is the integral part of the SDLC that deals with the test related phases of the software development.

* Requirement Analysis
* Test Planning
* Test case development
* Test Environment setup
* Test Execution
* Test Cycle closure
* **What is HP ALM (Quality Center) used for? Or What are the benefits and features of Quality Center?**

HP ALM (Application Life Cycle Management) is a tool that is used to manage, record, maintain and track

the SDLC activities from beginning to end efficiently as well as to provide detailed reports about the SDLC process, performance of the parties involved and time management.

**HP ALM;**

* Enables stakeholders to **interact and coordinate,** to achieve the project goals.
* Provides robust **tracking & reporting** and seamless integration of various project related tasks.
* Enables detailed **project analysis and effective management.**
* Can connect to email systems and send emails about any changes such as requirement change, defect raising, etc. to all desired team members.
* **What is the difference between TestDirector and Quality Center?**

Quality Center is upgraded version of Test Director that has enhanced Security, test and defect management features when compared to Test Director.

* **What is pair testing?**

Pair testing is a type of Adhoc testing where two testers are assigned on same module work on finding

defects through random approach. Usually one person executes the test while other taking notes on findings

* **What different types of testing may be considered and used for testing the web applications?**

1. Manual testing
2. Automation testing
3. Static testing
4. Dynamic testing
5. White box testing
6. Black box testing
7. Functional testing
   * + Unit testing
     + Integration testing
       - Smoke testing
       - Sanity testing
     + System testing
     + Interface testing
     + Regression testing
     + User acceptance testing
       - Alpha testing
       - Beta testing

* Exploratory testing
* Ad-hoc testing
  + Monkey testing
  + Buddy testing
  + Pair testing
* Positive testing
* Negative testing

1. Non-Functional testing

* Documentationtesting
* Installation testing
* Reliability testing
* Security testing
* Penetration testing
* Ethical hacking testing
* Performance testing
  + Load testing
  + Stress testing
  + Endurance testing
  + Spike testing
  + Volume testing
  + Scalability testing
* **What are the different types of software quality model?**

Software Quality Models are a standardized way of measuring the quality a software product. New applications are planned and developed every day proves the need for reassuring that the products-built meet or exceed the expected standards.

* 1. McCall Model
  2. Boehm Model
  3. FURPS Model
  4. Dromey Model
  5. ISO IEC 9126 Model
* **How to write a test case?**

**TEST CASE** is a set of actions executed to verify a particular feature or functionality of your software application works as expected. Writing good test cases requires basic writing skills, attention to detail, and a good understanding of the application under test (AUT).

Keeping in mind tester writing the test case can be different person than executing the tests,

**A good test case consists following:**

* **A strong title** that tells a clear message of what functionality is being tested
* **A strong description** that explains what needs to tested, the test environment, test data, preconditions, assumptions, dependencies on the test environment, and any special setup requirements. A good description will keep the test steps short and clear
* **A clear expected result**
* **Reusable test cases** provide long-term value to the software testing team. Testers can save time down the road by re-using the test case instead of re-writing it.
* **Why and how to prioritize test cases?**

Software quality can be assured by going through software testing process. However, software testing phase is an expensive process as it consumes a lot of time.

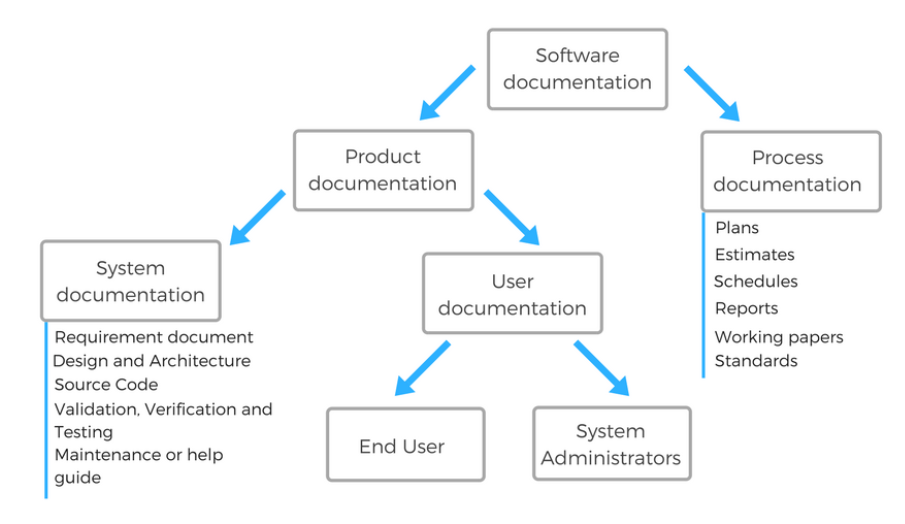
We run **regression testing** whenever there is a change to the software application to verify that change has not affected any old functionality and application is stable to carry out further testing. However, if we have thousands of test cases in regression suite and we do not have sufficient time to execute all. That’s why regression test suite has all the prioritized test cases covering major functionalities of the application

Test case prioritization is a method to prioritize and schedule test cases. The technique is developed in order to run test cases of higher priority in order to minimize time, cost and effort during software testing phase.

In order to **prioritize**, we need to categorize the test cases first

* **Blockers**: If feature stops working then this will block the further testing and issue has to get fixed on priority.
* **Critical**: These functionalities are very important to the customer and if these fail then customer will stop using the software so they also need to be fixed ASAP to avoid huge loss in business.
* **Major**: These functionalities make our software unique in the market and different from competitors. If these stops working, customer will be unhappy but will still use the software as all the critical features are working fine. They are also important after blockers and critical test cases as this can lead to loss in business as well.
* **Minor**: In this category, all the suggestions, and small UI changes or product improvements will be included. They will not affect the software usage in anyway and can be avoided if there is tight deadline.
* **List out different types of documentation/documents used in the software**

Software documentation is all written documents and materials related to software product’s development and use. Documentation exists in software development process to explain product functionality, unify project-related information, and allows stakeholders and development team to discuss all significant questions as well as providing a manual to end users.



* **How system testing is different to acceptance testing?**

|  |  |
| --- | --- |
| **System Testing** | **Acceptance Testing** |
| * System testing is to test end to end functionality of the software meets specified requirements | * Acceptance Testing is used to check whether the software meets the customer requirements or not * It is done after the system testing |
| * System testing is used by developers as well as testers | * Acceptance testing is used by testers, stakeholders as well as clients. |
| * Both functional and non-functional testing | * Only functional testing |
| * consists of System and integration testing | * Acceptance testing is the constitute of alpha and beta testing |
| * positive and negative test cases | * positive test cases |

* **List out different approaches and methods to design tests.**

The test case design techniques are classified into three major categories.

* + **Specification-Based (Black-Box) techniques**
    - Boundary Value Analysis
    - Equivalence Partitioning
    - Decision Table Testing
    - State Transition Diagrams
    - Use Case Testing
  + **Structure-Based (White-Box) techniques**
    - Statement Testing and Coverage
    - Decision testing coverage
    - Condition Testing
    - [Multiple Condition Testing](http://www.professionalqa.com/multiple-condition-coverage)
    - [All Path Testing](http://www.professionalqa.com/path-testing)
  + **Experience-Based techniques**
    - [Exploratory Testing](http://www.professionalqa.com/exploratory-testing)
    - [Error Guessing](http://www.professionalqa.com/error-guessing)
* **Whether test coverage and code coverage are similar terms?**

Code coverage and test coverage metrics are both to measure quality of your application code.

**Unit testing** is very important part of developing a high-quality software as developers find defects while writing codes. It avoids future defects, improves the quality and reduces delivery time and cost. It also provides developers a peace of mind that if they break a piece of code it will be found by the unit test.

**Code Coverage:** Percentage of codes that are covered **by automated unit tests**.(White-Box Testing Mentality)

(Statement coverage, condition coverage, branch coverage, toogle coverage, FSM coverage).

**Test Coverage:** Percentage of codes that are covered **by any type of tests.** (Black-Box Testing Mentality)

It helps determine whether the test cases are covering entire functional requirements.

**(**Such as unit tests, as well as functional tests, UI automation tests, performance tests, integration tests, system or acceptance tests.**)**

* **What a test report should contain?**
* **Project Info**

The title of your project, product name, and version in the test report.

* **Test Platform Details**

Details of every single platform and environment tested

* **Test Objective**

The objectives of each stage of software testing process such as functional testing, performance testing, UI testing etc.

* **Test Summary**

It is important to present this information visually by using color indicators, graphs, and highlighted tables.

* + The number of the executed test cases
  + The number of the passed test cases
  + The number of the failed test cases
  + Passed test cases percentage
  + Failed test cases percentage
  + Comments
* **Defects**
  + Total number of detected bugs
  + Bugs statuses (open, closed, fixed etc.)
  + Number of bugs by each status (open, closed, fixed etc.)
  + Severity and priority breakdowns
* **What are the roles and responsibilities of a tester or a QA engineer?**

**QA Engineer Roles**

* **Manual QA Engineer**
* **Automation QA Engineer**
* **QA Lead**

**A QA engineer is responsible to:**

* + Review requirements, specifications and technical design documents to provide timely and meaningful feedback
  + Create detailed, comprehensive and well-structured test plans and test cases
  + Estimate, prioritize, plan and coordinate testing activities
  + Design, develop and execute automation scripts using open source tools
  + Identify, record, document the bugs thoroughly and track bugs
  + Perform thorough regression testing when bugs are resolved
  + Develop and apply testing processes for new and existing products to meet client needs
* **What is positive and negative testing?**

**Positive Testing:** Verifying that software is behaving as it should for a valid data input.

**Negative Testing:** Verifying that software is behaving as it should for an invalid data input

* **When to start and stop testing?**

**When to Start Testing**

* Testing starts right from the requirements phase so that Requirements related defects caught later in the SDLC result in higher cost to fix the defect

**When to Stop Testing**

* Deadlines (release deadlines, testing deadlines, etc.)
* Test cases completed with certain percentage passed
* Test budget runs out
* Coverage of code/functionality/requirements reaches a specified point
* Bug rate falls below a certain level
* Beta or alpha testing period ends
* **Difference between load and stress testing.**

|  |  |
| --- | --- |
| **Load Testing** | **Stress Testing** |
| * Test the performance of the system or software application under extreme load. | * Test the robustness and stability of the system or software application under extreme load and data processing. |
| * High number of users. | * High number of users and too much data. |

* **Whether a software application can be 100% tested?**

**Exhaustive testing is not possible:** It is the process of testing the functionality of a software in all possible inputs (valid or invalid) and pre-conditions is known as exhaustive testing. Exhaustive testing is impossible means the software can never be tested at every test case.

* **Which is better approach to perform regression testing: manual or automation?**

**Regression Testing** is conducted to ensure that the new code changes has not affected the existing functionalities. Frequent code changes require frequent testing and this makes automation testing a better candidate for regression testing.

If automation of regression testing scenarios saves money without losing the quality of the product. Therefore, longer and critical scenarios should be run manually as there are chances of failure due to minor bugs in the system.

* **Does test strategy and test plan define the same purpose?**

[**Test Plan**](https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html) is a document which outlines the scope, objective, method and weight on a software testing task

[**Test Strategy**](https://www.guru99.com/how-to-create-test-strategy-document.html) is a set of principles that explain the test design and determine how testing needs to be done

* Test Plan is used at the project level whereas Test Strategy is used at the organization level.
* Test Plan has the primary goal of how to test, when to test and who will verify whereas Test Strategy has the primary goal of what technique to follow and which module to check.
* Test Plan can be changed whereas Test Strategy can’t change.
* Test Plan is carried out by the test manager whereas the Test Strategy is carried out by the project manager.
* **What is the difference between verification and validation approach of software testing?**

**Verification:** Ensuring the system is well-engineered, error-free

Are we building the system right?

Validation: Ensuring system meets customer requirements

Are we building the right system?

* **Why non-functional testing is equally important to functional testing?**

Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing for completeness of a software for satisfying a user experience.

* **What is the advantage of automation over manual testing approach and vice-versa?**

**Pros of Manual Testing:**

* Get fast and accurate visual feedback
* Less expensive as no required your budget for the automation tools and process however in the long run it may be more costly.
* Human judgment and intuition always benefit the manual element
* While testing a small change, an automation test would require coding which could be time-consuming. While you could test manually on the fly.

**Cons of Manual Testing:**

* Less reliable since conducted by a human. Always prone to mistakes & errors.
* The manual testing process can't be recorded and reused.
* Certain tasks are difficult to perform manually which may require an additional time of the software testing phase.

**Pros of automated testing:**

* Automated testing helps you to find more bugs compare to a human tester
* More speedy and efficient process
* Automation process can be recorded and reused
* Increases productivity because it provides fast & accurate testing result
* Testing coverage can be increased because of automation testing tool never forget to check even the smallest unit

**Cons of Automated Testing:**

* it's difficult to get insight into visual aspects of your UI like colors, font, sizes, contrast or button sizes.
* The tools to run automation testing can be expensive
* Every automation tool has their limitations which reduces the scope of automation.
* Debugging the test script is another major issue in the automated testing. Test maintenance is costly.
* **How to categorize bugs or defects found in the software?**
  + - **Critical:** A core functionality of the system fails or the system doesn’t work at all.
    - **Major:** The defect impacts basic functionality and the system is unable to function properly.
    - **Moderate**: The defect causes the system to generate false, inconsistent, or incomplete results.
    - **Minor:** The defect impacts the business but only in very few cases.
    - **Cosmetic:** The defect is only related to the interface and appearance of the application.
* **Difference between retesting and regression testing.**

Re-testing is to confirm the test cases that failed in the final execution are passing

after the defects are fixed

Regression Testing is to check whether a recent program or code change has not affected existing features

* **Smoke and Sanity testing are used to test software builds. Are they similar??**

[**Smoke Testing**](https://www.guru99.com/smoke-testing.html) is performed after initial builds to verify that the critical functionalities of the program are working fine. It is executed "before" any detailed functional or regression tests are executed on the software build. The purpose is to reject a badly broken application so that the QA team does not waste time installing and testing the software application. The objective is not to perform exhaustive testing

**Sanity testing** is performed after receiving a relatively stable software build, with minor changes in code, or functionality, to verify that the bugs have been fixed and no further issues are introduced due to these changes. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

* **If black-box and white-box, then why gray box testing?**
* In White Box testing -----------------------------internal structure (code) is known
* In Black Box testing ------------------------------internal structure (code) is unknown

**Grey Box Testing (combination of both) - internal structure (code) is partially known**

* 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
* **Which is a better testing methodology: black-box testing or white-box testing?**

In order to have a high-quality product which is bug free and no defect reaches the production. The Unit tests should be the first priority to find defects before defects reach to later stages of testing.

White-box testing is a testing technique which checks the internal functioning of the system. In this method, testing is based on coverage of code statements, branches, paths or conditions. White-Box testing is considered as low-level testing (Unit Test)

Black Box: 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.

So, First Priority should be White Box testing and then comes the Black Box testing.

* **Explain About the Software Maintenance Process?**

Software Maintenance is the process of modifying a software product after it has been delivered to the customer. The main purpose of software maintenance is to modify and update software application after delivery to correct faults and to improve performance.

* **Explain About** **the Software Release Process?**
* **What Is Low Level Design or Detailed Design?**

The view of the application developed during the HLD is broken into modules and programs. Logic design is done for every program and then documented as program specifications. For every program, a unit test plan is created. A good LLD makes developing easier.

Entry criteria is HLD document.

Exit criteria is Program Specifications and Unit Test Plan.

* **What Is High Level Design?**

HLD gives the overall system design in terms of functional architecture and database design. It designs the overall architecture of the entire system from main module to all sub module. This is very useful for developers to understand the flow of the system. In this phase design team, testers and customers plays a major role.

Entry criteria is SRS requirement document.

Exit criteria is HLD, Project Standards, Functional Design Documents, Database Design Document

* **What Is SRS?**

SRS-Software Requirement Specification is a document that describes what the software will do and how it will be expected to perform. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow. It’s used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help you make decisions about your product’s lifecycle — for instance, when to retire a feature. Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

* **What Is BRS?**

BRS - Business Requirement Specification is usually created at the very beginning of the product’s life cycle and describes the core product goals or needs client is willing to achieve with certain software or product. This one is usually created by a business analyst based on other stakeholders’ specifications and after a thorough analysis of the client company.

A BRS includes all the requirements requested by a client. Generally, it consists of the product’s purpose, users, the overall scope of work, all listed features and functions, usability and performance requirements. A BRS is used mainly by upper and middle management, product’s investors, business analysts.

* **What Is Scrum Methodology in Agile Software Development?**

Scrum is a framework in which teams solve complex problems and find high quality, innovative solutions for developing a software. Scrum structures developments into cycles of work that are called sprint under the principles of transparency, inspection and adaptation.

* **What Spiral Model?**
* **What is test execution?**

Test execution is the process of executing the code while checking the expected and actual results are the same to verify requirements and specifications of the software are met.

* **What are the different levels of testing?**
  + Unit Testing
  + Integration Testing
  + System Testing
  + Acceptance Testing
* **What is unit testing?**

Unit testing is testing the individual part of a software while coding to verify its working as expected to avoid future defects. It’s a white box testing done by developers.

* **What is the role of developer in unit testing?**

Unit testing is a white box testing done by developers. It also helps developers to understand the code base well and enables them to makes changes faster. It also avoids future defects and reduces delivery time and cost of the product.

* **What is integration testing?**

Integration testing is done to verify individually fine-working components of a software works as expected after integration. It mainly focuses on the interfaces and flow of the data between modules/units after integration. It can be white or black box testing.

* **What is system testing?**

System testing (end to end testing) is the process of testing a complete integrated system as a whole to ensure system functions and performs as expected. It can be white or black box testing.

* **What is acceptance testing?**

Acceptance testing is done to verify that software meets the business, customer requirements and user expectations before the release. Any issues found during acceptance testing are high priority issues that need to be fixed immediately.

* **Different types of acceptance testing.**
* Alpha testing
* Beta testing
* **Difference between alpha and beta testing.**

|  |  |
| --- | --- |
| **Alpha testing** | **Beta testing** |
| Alpha testing is done to verify defect-freeness and readiness of the software before release. | Beta testing is done to get feedback from users and to improve the software |
| It’s done internally by developers, testers and organization. | It’s done after a partial release externally to selected users. |
| It can be white or black box testing. | It’s a black box testing. |

* **What are the different approaches to perform software testing?**

**White Box Testing approach, Black Box Testing Approach, Gray Box Testing**

**White Box Testing** is testing the internal structure, design and coding of a software for verifying;

* The flow of inputs and outputs through the software
* Improving the design
* Usability and security

**Black Box Testing** is testing the functionality of the software without looking at the internal structure by verifying actual result is same as expected result.

**Gray Box Testing** is combination of both.

* **What Is Prototype Model?**
* **What Are the Advantages and Disadvantages of V Model?**
* **What Are the Advantages and Disadvantages of Waterfall Model?**

|  |  |
| --- | --- |
| **Advantages of Waterfall** | **Disadvantages of Waterfall** |
| Easy to understand and manage | Its inflexible and les iterative |
| Less production issues | Do not handle unexpected risk |
| Phases do not overlap | Its not suitable for complex and long-term projects |
| Better budget management | Difficult to make changes, requires going thru the same phases from beginning |
| Provides clear structure to the staff | Difficult to capture all the requirements upfront |

* **What Is Meant by Agile Model?**

Agile is a collection of values, principles and making project management decisions that are consistent with these principles.

Agile gives people a common foundation on decision making for developing software in best iterative way possible. It’s a feedback dependent methodology that includes customer in the process more than other methodologies.

* **What are test plan, test suite and test case?**

**Test Plan**: A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

**Test Suite:** A set of several test cases for a component or system under test, where the post condition of one test is often used as the precondition for the next one.

**Test Case** is a set of actions executed to verify a feature or functionality of a software works as expected. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement.

* **How to design test cases?**
* Use a Strong Title
* Include a Strong Description
* Include Assumptions and Preconditions
* Keep the Test Steps Clear and Concise
* Include the Expected result
* Make it Reusable
* **What is test environment?**

A testing environment is a setup of software and hardware for the testing teams to execute test cases.

* **Why test environment is needed?**

Setting up a right test environment ensures software testing success. It allows testers to test modules without affecting the production servers. Any flaws in this process may lead to extra cost and time to the client.

* **What Is Meant By Waterfall Model?**

Waterfall model is the first SDLC methodology originated from manufacturing. It’s a project management methodology based on sequential design process where one phase completely finished before starting on the next one. Waterfall is about quality, completeness and getting it right in the first time therefore, it requires thorough planning and documentation.

* **What Is Maintenance Phase?**

After software is up and running, it often requires continuous maintenance. In general, software remains operational for an extended period of time after initial implementation and requires regular maintenance to ensure that the software operates continually at peak performance levels. During the maintenance phase of the software life cycle, software programmers regularly issue software patches to address changes in the needs of an organization, to correct issues relating to bugs in the software or to resolve potential security issues. Throughout the maintenance phase, designers address issues that are discovered to prevent any hindrance to the expected performance of the software or to add increased functionality to the software.

* **What Is Deployment Phase?**

During this phase, release team executes the tasks such as;

* Installation of the system into production
* Installation of hardware and servers
* Setting up databases, links for software to go live to real users
* Writing user documentation
* Providing training
* **What Is Testing Phase?**

During this phase,

* Testers start wiring test conditions according to the requirements
* Performs testing with expected results
* Verifies system works as expected each and every time and meets customer requirements
* Find bugs and reports them to be fixed
* **What Is Coding Phase?**

During this phase, development team

* Builds the technical architecture,
* Build the database and programs then
* Developers start coding as well as tester start writing test cases
* **What Is Design Phase?**

During this phase, technical architecture gets designed

* Required software, hardware and telecommunication equipment,
* Programming languages, tools, frameworks,
* System server design, database relationship and
* Supported browser and platforms
* **What is the Difference Between CRS and SRS?**

|  |  |
| --- | --- |
| **CRS** | **SRS** |
| |  | | --- | | CRS stands for Customer Requirement Specification. | | SRS stands for System Requirement Specification. |
| It is Business Requirement Specification | It is Functional Specification |
| This document is provided by Customer, which contains detailed information about customer Business. | This document contains details about system modules and their functionality. |

* **What Is Requirement Gathering Phase?**

Requirements describe the features of the system need to be designed and developed. This is a most crucial phase in the software development lifecycle. Business Analyst will collect basic requirements from the clients like what does the software actually do, who are target customers, what should be achieved and how to achieve it.

After gathering the requirements, BA will prepare requirements document by adding all the customer needs. The requirements document then gets shared with client and gets approved by the customer. After approval, BA will give an overview for the team (team leads, designer, developers, tester) to accomplish end user requirement. All requirements need to be documented perfectly. So, it will be easy for the whole team to understand the client expectations

* **What Are the Models in SDLC?**
* Waterfall Model
* Agile Model
* Prototyping Model
* V-Shaped Model
* Rapid Application Model (RAD)
* Spiral Model
* Devops Model
* Joint Application Development (JAD)
* Iterative Model
* **What Are the Phases Of SDLC?**

1. Planning Phase
2. Requirement Analysis Phase
3. Design Phase
4. Development Phase
5. Testing Phase
6. Deployment Phase
7. Maintenance Phase

* **What is SDLC?**

SDLC is a process for developing a high-quality software that meets or exceeds customer expectations in a pre-defined time and cost.

* [**What is software testing life cycle (STLC)**](https://www.thinksys.com/qa-testing/complete-guide-to-stlc/)**?**

STLC is types of activities that are perform to ensure the quality of the product.

It is the integral part of the SDLC that deals with the test related phases of the software development.

* Requirement Analysis
* Test Planning
* Test case development
* Test Environment setup
* Test Execution
* Test Cycle closure
* **What is Software Quality Assurance (SQA)?**

SQA is a set of activities to ensure desired quality in software engineering process. It is a procedure of monitoring all the methods and activities of software development life cycle (SDLC). It focuses on preventing defects or flaws in the wholesome software development process and in addition to this, it also ensures whether all the approaches, techniques, methods, and processes are implemented correctly or not while execution of SDLC. It is done by continuous comparing of the current procedures with the appropriate and well-defined standards.

* **What is Software Quality Control (SQC)?**

SQC is a method of maintaining and achieving the quality standards in Software Products with the assistance of testing against the predefined, standard specification. SQC is a reactive and corrective procedure through which an undeveloped product grows into the end product.

* **What is Software Testing?**

**Software Testing** is the process of evaluating a system or its components with the intent to find whether it works as expected and satisfies the specified requirements or not.

Testing is a subset of SQC. Testing is carried out to identify the bugs present in the software product. The bugs are passed to the developers, who then fix them. After debugging, the product is verified once again, in order to have the product as per the client expectations.

* **Whether, Software Quality Assurance (SQA), Software Quality Control (SQC) and Software Testing are similar terms?**

Software Quality Assurance (SQA), Software Quality Control (SQC) and Testing are related but not the same.

* **Then, what’s the difference between SQA, SQC and Testing?**

|  |  |
| --- | --- |
| **Software Quality Assurance** | **Software Quality Control** |
| SQA ensures quality in software engineering processes which establish and evaluate the processes that produce products. | SQC ensures quality in software products which focuses on identifying defects in the actual products produced |
| Goal is to prevent the defect | Goal is to identify and improve the defects |
| It is the technique of managing the quality | It is a method to verify the quality. |
| Not an execution process | Execution process |
| Prevention oriented | Detection oriented |
| Process focused | Product focused |
| Assures you what are doing is the right things. | It assures that the end result of what you had done is exactly what you expected. |
| Prior and during the SDLC | During the STLC |

Testing is a subset of SQC. Testing is carried out to identify the bugs present in the software product.

* **How STLC is related to or different from SDLC (software development life cycle)?**

STLC is the integral part of the SDLC that deals with the test related phases of the software development.

* **What are the phases involved in the software testing life cycle?**
  + - Requirement Analysis
    - Test Planning
    - Test case development
    - Test Environment setup
    - Test Execution
    - Test Cycle closure
* **Why entry criteria and exit criteria is specified and defined?**

A [**software testing process**](http://www.professionalqa.com/test-process) if not bound by any stipulation or criteria, it may be initiated at any time and may be carried out for the infinite time or for a very short duration. This may produce negative consequences such as absence of goals and objectives, unable to meet the deadlines, over-budget, inefficient testing due to inadequate requirements and understanding of the product.

A QA team needs to know and understand that when it is suitable to carry out the testing process and at what point the process needs to be terminated. It should be known where to start and where to stop. In short, a testing process should be defined and planned along with its **entry and exit criteria**.

**Entry criteria** are the condition or the set of conditions, which should exist or be met in order to start a process.

**Exit Criteria** is the condition of the set of conditions which imparts the completion of an activity or meeting of the targets and goals.

* **What are the different types of requirements required in software testing?**
* Business Requirements
* System Requirements
  + Functional Requirements (FR)
  + Non-Functional Requirements (NFR)
* UI Requirements (UIR)
* **Why there is a bug/defect in software?**

A Software Bug is a failure or flaw in a program that produces undesired or incorrect results. It’s an error that prevents the application from functioning as it should. There are many reasons for Software Bugs.

* + - **Human factor:**
    - **Communication failure**
    - **Unrealistic development timeframe**
    - **Poor design logic**
    - **Poor coding practices**
    - **Lack of version control**
    - **Buggy third-party tools**
    - **Lack of skilled testing**
    - **Last minute changes**

What is build and release?

A “**build**” is given by dev team to the test team. A “**release**” is formal **release** of the product to its customers. A **build** when tested and certified by the test team is given to the customers as “**release**”. A “**build**” can be rejected by test team if any of the tests fail or it does not meet certain requirements.

**Topic : Database – MYSQL – Practical & Theory**

### **In which language MySQL has been written?**

MySQL is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B)

* **What is SQL?**

SQL is structural query language. SQL is the standard language for interacting with Relational Database Management Systems.

* **What are the differences between DDL, DML and DCL in SQL?**

DDL, DML and DCL are subsets of SQL.

* + **DDL - Data Definition Language**

Create, Alter, Drop, Rename, Truncate, Comment

* **DML - Data Manipulation Language**

Select, Insert, Update, Delete, Merge, Call, Explain Plan, Lock Table

* **DCL - Data Control Language**

Grant, Revoke

* **What is a Database?**

Database is an organized collection of structured information or data

* **Does SQL support programming language features?**

SQL is a language but it does not support programming. it is a command language not a programming language. SQL does not have conditional statements like for loops or if, else. Only commands which we can use to query, update, delete etc. data in the database. SQL allows us to manipulate data in a database.

* **What are the differences between SQL and PL/SQL?**

|  |  |
| --- | --- |
| **SQL** | **PL/SQL** |
| SQL is a query execution or commanding language | PL/SQL is a complete programming language |
| SQL is data-oriented language | PL/SQL is a procedural language |
| It is used for manipulating data | It is used for creating applications |
| We can execute one statement at a time in SQL | We can execute block of statements in PL/SQL |
| SQL tells database, what to do? | PL/SQL tells database how to do |

* **Write an SQL query to find names of employee start with ‘A’?**

Select first\_name, last\_name from employees

Where upper (first\_name) like ‘A%’;

* **What is an index?**

It is a data structure technique which is used to quickly locate and access the data in a database. An index is created on a column of a table and data structure (most commonly a B- tree) that stores the values for a specific column in a table.

### **What are the technical specifications of MySQL?**

### **What is the difference between MySQL and SQL?**

MYSQL is a relational database management system that uses SQL to create, maintain and manage the date in a secure way.

SQL is structured query language that allows us interact with RDBMS’.

### **What is the difference between database and table?**

**Database** is a collection of several components like **tables**, indexes, stored procedures and so on. **Table** is a two-dimensional structure that contains data in rows and columns where is row is a specific record and each column is a specific field for data to be stored.

### **What are the disadvantages of MySQL?**

* MySQL does not support a very large database size as efficiently.
* MySQL does not support ROLE, COMMIT, and Stored procedures in versions less than 5.0.
* Transactions are not handled very efficiently.
* There are a few stability issues.
* It suffers from poor performance scaling.
* The development is not community driven so it has lagged behind.
* The functionality tends to be heavily dependent on the addons.
* Developers may find some of its limitations very frustrating.
* **Write a SQL statement to create a simple table countries including columns country\_id, country\_name and region\_id.**

CREATE TABLE countries (

country\_id INT primary key,

country\_name VARCHAR(40),

region\_id INT);

**Project & Practical: Manual Testing, Manual Web Testing & Manual Hardware Testing**

* **Scenario 1:**

**You work as a Sr. Software Test Engineer at Accenture. Your QA manager, John Smith, has asked you to take the lead on an upcoming project in the QA Pipeline. As part of your responsibilities, you’ve designed a Test Plan and submitted it for approval to John. John has extensive leadership experience in the IT industry and has managed other areas of IT previously. However, John hasn’t had much experience in the Quality Assurance world. After reviewing your Test Plan, John asks for some clarification on some of the sections you’ve created in your Test Plan. Write an Email to John providing more details on the sections he’s asked about (listed below)**

* **Test Scope (Features to be tested and Features not to be Tested)**
* **Pass/Fail Criteria**
* **Entry/Exit Criteria**
* **Test Deliverables**
* **Environmental Needs**
* **Responsibilities**

**Be sure to provide an explanation for each of the sections mentioned above in your email. Your email should be professional and succinct.**

## Scenario 2:

**You are a Software QA Professional and are now interviewing at Amazon for your next job. While in the interview, the QA Lead taking your interview asks you to explain the difference between Regression Testing and Adhoc testing. Provide a thorough and complete answer to this interview question.**

Adhoc testing is an informal testing type with an aim to break the system and find defects by random checking. Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application. Ad hoc testing can be performed when there is limited time to do elaborative testing. Usually adhoc testing is performed after the formal test execution. And if time permits, ad hoc testing can be done on the system. Ad hoc testing will be effective only if the tester is knowledgeable of the System Under Test by guessing the most likely source of errors.

We run **regression testing** whenever there is a change to the software application to verify that change has not affected any old functionality and application is stable to carry out further testing. Test case prioritization is a method to prioritize and schedule test cases. The technique is developed in order to run test cases of higher priority in order to minimize time, cost and effort during software testing phase.

* **Test Scenario 3: Test Scenarios of ATM Machine (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ID\_atm\_001 | Verify user can insert a card to the atm machine |
| TC\_ID\_atm\_002 | Verify user can login with valid pin number |
| TC\_ ID\_atm\_003 | Verify user cannot login with an invalid pin number |
| TC\_ ID\_atm\_004 | Verify user can withdraw amount of money that’s smaller than the balance |
| TC\_ ID\_atm\_005 | Verify user can deposit money |

* **Test Scenario 4: Test scenarios of Bike (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_bike\_001 | Verify bike can carry up to 300 lbs |
| TC\_ ID\_bike\_002 | Verify rims dimensions are correct |
| TC\_ ID\_bike\_003 | Verify bike stops when brake |
| TC\_ ID\_bike\_004 | Verify bike moves when pedal |
| TC\_ ID\_bike\_005 | Verify bike seat can be adjusted for height |

* **Test Scenario 5: Test Scenarios of Calculator (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_calculator\_001 | Verify user can turn on the calculator |
| TC\_ ID\_calculator\_002 | Verify user can enter numbers to the calculator |
| TC\_ ID\_calculator\_003 | Verify user can use divide function |
| TC\_ ID\_calculator\_004 | Verify user can use multiply function |
| TC\_ ID\_calculator\_005 | Verify user can turn of the calculator |

* **Test Scenario 6: Test Scenarios of Car (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_car\_001 | Verify user can unlock the car with valid key present |
| TC\_ ID\_car\_002 | Verify user can adjust the driver seat |
| TC\_ ID\_car\_003 | Verify user can roll down all four windows |
| TC\_ ID\_car\_004 | Verify user can start the car |
| TC\_ ID\_car\_005 | Verify user can turn the radio on |

* **Test Scenario 7: Test Scenarios for Chair (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_chair\_001 | Verify chair has two or four legs |
| TC\_ ID\_chair\_002 | Verify chair can carry up to 300 lbs |
| TC\_ ID\_chair\_003 | Verify chair seat height is as specified |
| TC\_ ID\_chair\_004 | Verify chair seat size is as specified |
| TC\_ ID\_chair\_005 | Verify back support of the chair can support specified force |

* **Test Scenario 8: Test cases of Coffee Machine (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_coffeemach\_001 | Verify coffee machine turns on when plugged |
| TC\_ ID\_coffeemach\_002 | Verify coffee machine water reservoir volume as specified |
| TC\_ ID\_coffeemach\_003 | Verify coffee machine boils the water |
| TC\_ ID\_coffeemach\_004 | Verify coffee machine provides cup size options and cup size can be selected |
| TC\_ ID\_coffeemach\_005 | Verify coffee machine brews the coffee into the cup |

* **Test Scenario 9: Test Scenarios of Date field (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_datefield\_001 | Verify user can view the date field of the laptop |
| TC\_ ID\_datefield\_002 | Verify date field shows time |
| TC\_ ID\_datefield\_003 | Verify date field gets updated correctly as the time and day change |
| TC\_ ID\_datefield\_004 | Verify date and time can be changed |
| TC\_ ID\_datefield\_005 | Verify date field shows the day when bringing the pointer on |

* **Test Scenario 10: Test cases of Door (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_door\_001 | Verify door dimensions are as specified |
| TC\_ ID\_door\_002 | Verify door can handle functions as it should |
| TC\_ ID\_door\_003 | Verify door can be opened and closed |
| TC\_ ID\_door\_004 | Verify door materiel is as specified |
| TC\_ ID\_door\_005 | Verify door can be locked and unlocked |

* **Test Scenario 11: Test cases for Online Shopping Application (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_onlineshop\_001 | Verify user can launch the application |
| TC\_ ID\_onlineshop\_002 | Verify user can login to the application with valid username and password |
| TC\_ ID\_onlineshop\_003 | Verify user cannot login without valid username and password |
| TC\_ ID\_onlineshop\_004 | Verify user can search items |
| TC\_ ID\_onlineshop\_005 | Verify user can add items to the cart |

* **Test Scenario 12: Test Scenarios of Facebook (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_facebook\_001 | Verify user can send friend request |
| TC\_ ID\_facebook\_002 | Verify user can search other users |
| TC\_ ID\_facebook\_003 | Verify user can view the marketplace |
| TC\_ ID\_facebook\_004 | Verify user can post items to the market place |
| TC\_ ID\_facebook\_005 | Verify user can add photos |

* **Test Scenario 13: Test scenarios for Fan (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_fan\_001 | Verify fan can be turned on and off |
| TC\_ ID\_fan\_002 | Verify fan speed can be adjusted |
| TC\_ ID\_fan\_003 | Verify fan can work up specified time without any issues |
| TC\_ ID\_fan\_004 | Verify fan has a protection screen for safety |
| TC\_ ID\_fan\_005 | Verify fan can blow air up to 180-degree radius |

* **Test Scenario 14: Test Scenarios of Flight Reservation (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_flight\_001 | Verify user can enter destination airport |
| TC\_ ID\_flight\_002 | Verify user can enter origin airport |
| TC\_ ID\_flight\_003 | Verify user can search flight options |
| TC\_ ID\_flight\_004 | Verify user can view flight details |
| TC\_ ID\_flight\_005 | Verify user can book the flight |

* **Test Scenario 15: Test Scenarios of Gmail (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_gmail\_001 | Verify user can login successfully with valid username and password |
| TC\_ ID\_gmail\_002 | Verify user can view emails |
| TC\_ ID\_gmail\_003 | Verify user can mark all the emails as read |
| TC\_ ID\_gmail\_004 | Verify user can delete selected emails |
| TC\_ ID\_gmail\_005 | Verify user can send an email successfully |

* **Test Scenario 16: Test scenarios of Google Search (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_google\_001 | Verify user can launch ‘https://www.google.com’ |
| TC\_ ID\_google\_002 | Verify user can enter keywords in the search bar |
| TC\_ ID\_google\_003 | Verify user can view the search results |
| TC\_ ID\_google\_004 | Verify user can select a result and open the result page selected |
| TC\_ ID\_google\_005 | Verify user can view how many results found related to keywords |

## Scenario 17 :

**You are a Software QA Professional and are now interviewing at Google for your next job. While in the interview, the QA Manager taking your interview asks you the following question: “Let’s say you log a bug that you find while executing 1 of your tests cases. But the developer replies back and says that this is not a valid bug. How would you tackle this situation and what would be your next steps?” Provide a thorough and complete answer to this interview question.**

* **Scenario 18: Test Scenarios of WhatsApp (Minimum 5 Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_whatsapp\_001 | Verify user can launch WhatsApp |
| TC\_ ID\_whatsapp\_002 | Verify user can view the chat list |
| TC\_ ID\_whatsapp\_003 | Verify user can send a message |
| TC\_ ID\_whatsapp\_004 | Verify user can start a group |
| TC\_ ID\_whatsapp\_005 | Verify user can delete messages |

* **Scenario 19: Test Scenarios of YouTube (**Minimum5 **Test Cases in Excel Sheet) (No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_youtube\_001 | Verify user can launch ‘https://www.youtube.com’ |
| TC\_ ID\_youtube\_002 | Verify user can view the homepage |
| TC\_ ID\_youtube\_003 | Verify user can search videos |
| TC\_ ID\_youtube\_004 | Verify user can open a video |
| TC\_ ID\_youtube\_005 | Verify user can change a video quality |

* **Scenario 20 : Test cases of TV Remote Control (Minimum 5 Test Cases in Excel Sheet)(No Test Steps)**

|  |  |
| --- | --- |
| TC\_ ID\_TVremote\_001 | Verify user can turn on the TV from remote controller |
| TC\_ ID\_TVremote\_002 | Verify user can change channels from remote controller |
| TC\_ ID\_TVremote\_003 | Verify user can adjust the volume from remote controller |
| TC\_ ID\_TVremote\_004 | Verify user can change picture quality from remote controller |
| TC\_ ID\_TVremote\_005 | Verify user can voice search from remote controller |

**SUBMISSION INSTRUCTIONS:**

**ONLY 1 EMAIL MUST BE SENT TO** https://mail.google.com/mail/u/0/images/cleardot.gifhomework@peoplentech.com

**Save all your responses in a MS Word document and label the file using the following format: “Ankur Jain – Fall20 Bootcamp – Last 6 digits of Student ID” Example: Fall18 SDLC Bootcamp – 0501VA (0501VA is Jay’s Student ID)**

**Email :** [**Homework@peoplentech.com**](mailto:Homework@peoplentech.com)