

Module – 2 (Manual Testing) – Part: 2

- Que: 31. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?
- Que: 32. What is the difference between test scenarios, test cases, and test script?
- Que: 33. Explain what Test Plan is? What is the information that should be covered.
- Que: 34. What is priority?
- Que: 35. What is severity?
- Que: 36. Advantage of Bugzilla.
- Que: 37. Difference between priority and severity
- Que: 38. What are the different Methodologies in Agile Development Model?
- Que: 39. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?
- Que: 40. What is the procedure for GUI Testing?

Que: 31. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

No.	SDLC	STLC
1.	SDLC is mainly related to software development.	STLC is mainly related to software testing.
2.	Besides development other phases like testing is also included.	It focuses only on testing the software.
3.	SDLC involves total six phases or steps.	STLC involves only five phases or steps.
4.	In SDLC, a greater number of members (developers) are required for the whole process.	In STLC, a smaller number of members (testers) are needed.
5.	In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team(Test Lead or Test Architect) makes the plans and designs.
6.	Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software
7.	It helps in developing good quality software.	It helps in making the software defects free.
8.	SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
9.	Post deployment support , enhancement , and update are to be included if necessary.	Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.
10.	Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC.

Que: 32. What is the difference between test scenarios, test cases, and test script?

No.	Test Scenario	Test cases	Test Script
1.	It is any functionality that can be change	Is a set of action executed to verify particular features or functionality	Is a set of instruction to test to any app automatically
2.	Is derived from test artifacts like Business Requirement Specification(BRS) and Software Requirement Specification(SRS)	Is Mostly derived from test scenarios.	Is mostly derived from test cases.
3.	Helps test the end-to-end functionality in an Agile way	Helps in exhaustive testing of an app	Helps to test specific things repeatedly
4.	Is more focused on what to test	Is focused on what to test and how to test	Is focused on the expected result
5.	Takes less time and fewer resources to create	Requires more resources and time	Requires less then time for testing but more resources for scripts creating and updating
6.	Includes an end-to end functionality to be tested	Includes test steps, data, expected result for testing	Includes different commands to develop a script
7.	The main task is to check the full functionality of a software application	The main task is to verify compliance with the applicable standards, guidelines, and customer requirements.	The main task is to verify that nothing is skipped, and the results are true as the desired testing plan
8.	Allows quickly assessing the testing scope	Allows detecting errors and defects	Allows carrying out an automatic execution of test cases

Que: 33. Explain what Test Plan is? What is the information that should be covered.

A document describing the scope, approach, resources and schedule of intended test activities. Determining the scope and risks, and identifying the objectives of testing. Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

Test Planning Activities:

- **Approach:** Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.
- **Integrating and coordinating the testing activities into the software life cycle activities:** acquisition, supply, development, operation and maintenance. Making decisions about:
 - **what to test**
 - **Who does test?** i.e. what roles will perform the test activities
 - **when and how** the test activities should be done and when they should be stopped (exit criteria – see next slides)
 - **How the test** results will be evaluated Assigning resources for the different tasks defined.
- **Test ware:** Defining the amount, level of detail, structure and templates for the test documentation. Selecting metrics for monitoring and controlling test preparation and execution, defect resolution and risk issues.
- **Process:** Setting the level of detail for test procedures in order to provide enough information to support reproducible test preparation and execution.

Que: 34. What is priority?

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

For example: If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

- **Priority can be of following types:**
 - **Low:** The defect is an irritant which should be repaired, but repair can be deferred until after more serious defect has been fixed.
 - **Medium:** The defect should be resolved in the normal course of development activities. It can wait until a new build or version is created.
 - **High:** The defect must be resolved as soon as possible because the defect is affecting the application or the product severely. The system cannot be used until the repair has been done.
 - **Critical:** Extremely urgent, resolve immediately

- **What is severity?**

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

For example: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

Severity can be of following types: Critical, Major (High), Moderate (Medium), Minor (Low), Cosmetic.

Que: 36. Advantage of Bugzilla.

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database.

Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Centre, Testlink etc.

- Open source, free bug tracking tool.
- Automatic Duplicate Bug Detection.
- Search option with advanced features.
- File/Modify Bugs by Email.
- Move Bugs Between Installs.
- Multiple Authentication Methods (LDAP, Apache server).
- Time Tracking.
- Automated bug reporting; has an API to interact with system.
- Integrated email capabilities.
- Detailed permissions system.
- Optimized database structure to enhance performance.
- Robust security.
- Powerful query tool.
- Ideal for small projects.

Que: 38. What are the different Methodologies in Agile Development Model?

What is Agile?

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams' cycle through a process of planning, executing, and evaluating.

Agile is based on the iterative-incremental model. In an incremental model, we create the system in increments, where each increment is developed and tested individually.

Agile four manifesto:

- Individuals and interactions, Over processes and tools
- Working software, over comprehensive documentation
- Customer collaboration, Over contract negotiation
- Responding to change, over following a plan

Agile Principles:

- Customer satisfaction through early and continuous software delivery
Accommodate changing requirements throughout the development process
Frequent delivery of working software
- Collaboration between the business stakeholders and developers throughout the Project
- Support, trust, and motivate the people involved
- Enable face-to-face interactions
- Working software is the primary measure of progress
- Agile processes to support a consistent development pace
- Attention to technical detail and design enhances agility
- Simplicity
- Self-organizing teams encourage great architectures, requirements, and designs
- Regular reflections on how to become more effective

Scrum

Scrum:

SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team-based development environment. It consists of three roles and their responsibilities are explained as follows

- **Scrum Master:** Master is responsible for setting up the team, sprint meeting and removes obstacles to progress
- **Product owner:** The Product Owner creates product backlog, prioritizes the backlog and is responsible for the delivery of the functionality at each iteration
- **Scrum Team:** Team manages its own work and organizes the work to complete the sprint or cycle.
- **Sprint:** Sprint is a time-boxed period in which the scrum team needs to finish the set amount of work. Each sprint has a specified timeline, i.e., 2 weeks to 1 month. The scrum team agrees with this timeline during the sprint planning meeting.

Scrum Roles:

Product Owner:

There is a client who wants to develop his software, so he approaches to the company who can develop his software. What does the company do? The Company assigns a role, i.e., Product Owner. Product Owner is the person who communicates with the clients understands their requirements. Product Owner is the responsible person from the company for software development.

Scrum Master

During the sprint, Agile says that the team should meet together once daily. When the team is following scrum means that they are conducting meetings daily for 10 to 15 minutes. This meeting is known as a scrum meeting. Scrum Master is the person who handles the scrum meeting.

Scrum Team:

The team comprises of persons who work on the project. It can be developers, testers or designers. When we talk about Agile or Scrum then we talk about the team, we do not talk about developers, or testers as an individual. Agile says that developers can work as a tester or testers can work as a developer when the need arises.

Scrum Board

Product Backlog: Product Backlog is a set of activities that need to be done to develop the software. **Sprint Backlog:** Sprint Backlog is a backlog that has taken some of the activities from the product backlog which needs to be completed within this sprint.

Scrum Board: Scrum Board is a board that shows the status of all the activities that need to be done within this sprint. Scrum Board consists of four status:

Open: The 'Open' status means that the tasks which are available in 'Open' are not yet started.

In progress: The 'In progress' status means the developers completed their tasks.

Testing: The 'testing' means that the task is in a testing phase.

Closed: The 'closed' means the task has been completed.

Que: 39.Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

No.	Authentication	Authorization
1.	Authentication verifies who the user is.	Authorization determines what resources user can access.
2.	Authentication work through passwords, one time pin, biometric information, and other information provided or entered by the user.	Authorization work through settings that are implemented and maintained by the organization.
3.	Authentication is the first step of a good indent and access management process.	Authorization always take place after Authorization.
4.	Authentication is visible to and partially changeable by the user.	Authorization isn't visible to or not changeable by the user.
5.	Example: By verifying their identity, employees can again access to a human resource (HR) application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different level of data based on the permissions set by the organization.

What are the common problems faced in Web testing?

1) Insufficient testing for browser compatibility: Poor browser compatibility is an obvious turn-off – if users can't see your content clearly and transactional features don't work well, they'll move on.

2) Failing to conduct thorough functional testing across mobile: Many websites and apps suffer from functional issues on different mobile devices, because they haven't been tested on all the devices their customers are using.

3) Failing to conduct thorough functional testing across desktop: Yes, it's the same problem but in reverse. Some companies are so carried away with the news that mobile is predominant that they drop the ball on desktop testing. Responsive mobile design has become the default for many development teams. But although desktop usage may be in the minority, crucial segments of your target audience may over-index on desktop. Some users prefer browsing and reading on a larger screen.

4) Poor data security: Data security is a constant challenge for organizations to deal with. Hackers regularly probe company security for weaknesses. New viruses and malware are constantly being developed to exploit unlatched software vulnerabilities. Paying insufficient attention to security in your QA testing is a big risk.

5) Failing to provide an intuitive experience: While it's important that your website's visual design is up-to-date with the latest trends, a Hub Spot report reveals why a large number of consumers view easy navigation as the most important factor in website design. Common problems that can negatively affect website navigation are:

- Broken links
- Slow loading speeds
- Complicated menu structures

6) Not performing testing frequently enough: Tick off your pre-launch web testing and you can tick off any future problems right? Of course, this is not the case. You need to repeat website testing regularly to identify any conversion blockers that are costing you revenue, as well as any potential issues that appear over time before they become critical.

7) Leaving digital accessibility to the last minute: If you don't take digital accessibility seriously when you're creating websites and apps, you're missing out on a share of the £249 billion spent annually by customers with disabilities and impairments that can affect the way they engage with digital products.

8) Releasing new features breaks the existing live system: After any deployment or release, there's a risk of unintended consequences. Changes can introduce defects or alter the behaviour of a flow or specific function.

9) Localization and the global experience: Launching digital products to a worldwide audience, rather than one geography or in a single language, brings a whole lot more complexity. Customers and users of your software or website may interact in a language different from the one it was written in. Automated translations have come a long way, but they still don't pick up on the subtleties of language.

10) The most common bugs: Bugs in code and errors in set-up can ruin website or app user experiences. As well as getting in the way of smooth navigation and stopping users completing transactions, they give an impression of carelessness or a lack of attention to detail.

Que: 40. What is the procedure for GUI Testing?

Manual Testing: This approach involves human tester, where each screen is manually checked to validate each functionality by creating and executing test cases. It is a useful approach when part of UI or a feature is ready, the probability of defects is more at the initial stage, and human intervention is required.

Record and Replay Testing: GUI record and replay tools are used to test applications for their user interface. Using such tools, testers run an application and record the user interaction with the app. A script runs to track and save the user actions, including cursor movements, which can be replayed several times to find the issues in the interface.

Model-based testing: In this type of GUI testing, a model is created to understand and evaluate the system's behavior. This approach is useful in creating accurate test cases using system requirements. It is a structured, thorough, measurable form of testing.

There are three essential aspects of model-based GUI testing:

- Automatically generated test cases from the model
- Manually derived test cases from the model
- Model and requirements coverage metrics

Things to consider for model-based testing:

- Create the model
- Determine the information as inputs in the system
- Verifying the expected output
- Execute tests
- Checking and validating actual vs. expected
- Take further action on the model