MANUAL TESTING

1. **what is SDLC and different phases in SDLC?**

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

* Requirement gathering and analysis
* Design
* Implementation or coding
* Testing
* Deployment
* Maintenance

1. **what is the process in agile model**

Agile methodology is an alternative to traditional project management, typically used in software development. Agile processes generally promote a disciplined project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices intended to allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

1. what is scrum methodology

Scrum is an **iterative** and incremental **agile software development** methodology for managing product development. Scrum is part of the **Agile** movement.

1. what is daily standup meeting and what we discuss

A daily stand-up meeting is a short organizational meeting that is held each day. The meeting, generally limited to between five and fifteen minutes long, is sometimes referred to as a stand-up, a morning roll-call or a daily scrum.

The purpose of the meeting is for each team member to answer the following three questions:

1) What did you do yesterday?

2) What will you do today?

3) Are there any impediments in your way?

1. what is user story and task in user story

A **user story** is a tool used in Agile software development to capture a description of a software feature from an end-**user**perspective. The **user story** describes the type of **user**, what they want and why. A **user story** helps to create a simplified description of a requirement.

A user story is typically functionality that will be visible to end users. Developing it will usually involve a programmer and tester, perhaps a user interface designer or analyst, perhaps a database designer, or others. Cannot be developed by single person.

A task, on the other hand, is typically something like code this, design that, create test data for such-and-such, automate that, and so on. These tend to be things done by one person.

1. what is sprint planning and spring retro

The **sprint planning** is a one-day timeboxed meeting divided into two 4-hour sessions, which is also timeboxed. **Sprint planning** is a collaborative effort involving: ScrumMaster – to facilitate the meeting. Product Owner – to clarify the details of the product backlog items and their respective acceptance criteria. Entire **Agile** Team –to define the work and effort necessary to meet their commitment to complete product backlog.

1. what is burndown chart and velocity

The burndown is a chart that shows how quickly you and your team are burning through your customer's [user stories](http://www.agilenutshell.com/user_stories). It shows the total effort against the amount of work we deliver each iteration.

Velocity is not an indicator of the team. *Velocity is not a KPI*by which you should measure your team*.* *Velocity is just capacity planning tool*. Nothing more, nothing less.

1. what is product backlog item and sprint backlog items

Product Backlog Items (PBIs) are the elements that make up the Product Backlog. Product Backlog Items can range from specifications and requirements, to use cases, epics, [User Stories](http://www.scruminc.com/independent-user-stories/), or even bugs and chores. **Upon Completion you will:**

* Know what kinds of items can be found in a Backlog
* Have a checklist for creating a Product Backlog Item.

A Sprint Backlog Item can be seen as a Product Backlog Item plus a plan for implementing it. In that sense it is additive to, and thus "larger" than, the selected PBI. The sprint backlog is a list of tasks identified by the Scrum team to be completed during the [Scrum](https://www.mountaingoatsoftware.com/agile/scrum) sprint.

1. What is waterfall/v model/agile in SDLC?

Waterfall model is the Sequential development model. • Requirement should be clear before going to next phase of design.

The V model (Validation & Verification model)[11] is a modified version of the Waterfall method As opposed to the Waterfall method, this one was not designed in a linear axis; instead the stages turn back upwards after the coding phase is done. • This developmental process is balanced and relies on the verification from the previous steps before proceeding forward.

The term agile stands for 'moving quickly' • Agile methodology [12] has an adaptive team which is able to respond to the changing requirements. • Customer satisfaction by rapid delivery of useful software.

1. What is STLC?

Software Testing Life Cycle refers to a testing process which has specific steps to be executed in a definite sequence to ensure that the quality goals have been met. In STLC process, each activity is carried out in a planned and systematic way. Requirements phase

Planning Phase

Analysis phase

Design Phase

Implementation Phase

Execution Phase

Conclusion Phase

Closure Phase

1. What is defect?

A defect is an error or a bug, in the application which is created. A programmer while designing and building the software can make mistakes or error. These mistakes or errors mean that there are flaws in the software. These are called defects.

1. How to arise a defect and what we specify while logging defect?

Report a defect if there is a difference between the expected result and the actual result, unless the defect has already been reported.

1. Defect lifecycle

Defect life cycle, also known as Bug Life cycle is the journey of a defect cycle, which a defect goes through during its lifetime. It varies from organization to organization and also from project to project as it is governed by the software testing process and also depends upon the tools used.

1. Different types of testing?

###### 1.Build Acceptance Testing Or Build Verification Testing Or Sanity Testing: -

2.Regression Testing: -

3.Re-Testing: -

4. α- Testing (Alpha Testing): -

5. β-Testing (Beta): -

6. Static Testing: -

7. Dynamic Testing: -

8.Installation Testing: -

9.Compatibility Testing: -

10. Monkey Testing: -

11.Usability Testing: -

12. Exploratory Testing: -

13.End-To- End Testing: -

14.Port Testing:-

15.Reliability Testing: -

16.Security Testing: -

Authentication Testing: -

Direct URL Testing: -

Firewall Leakage Testing: -

17. Mutation Testing: -

18. ADHOC Testing: -

1. when do we use regression testing?

It is a type of testing in which one will perform testing on the already tested functionalities again and again.

It is usually done in two scenarios.

1. Whenever the testers has raised the defects, rectified by the developers and next build is released to the testing department then the test engineer’s will test the defect functionality as well as the related functionality once again.
2. When ever some new features are incorporated by the developers, next build is released to the testing department then the test engineers will once again test the related functionalities of the new features in order to confirm that they are working same as previous.

Note: - Testing the new functionalities for the first time is known as new testing but not Regression Testing.

1. when do we use integration testing?

**Integration testing** (sometimes called **integration and testing**, abbreviated **I&T**) is the phase in [software testing](https://en.wikipedia.org/wiki/Software_testing) in which individual software modules are combined and tested as a group. It occurs after [unit testing](https://en.wikipedia.org/wiki/Unit_testing) and before [validation testing](https://en.wikipedia.org/wiki/Verification_and_validation_(software)). Integration testing takes as its input [modules](https://en.wikipedia.org/wiki/Module_(programming)) that have been [unit tested](https://en.wikipedia.org/wiki/Unit_testing), groups them in larger aggregates, applies tests defined in an integration [test plan](https://en.wikipedia.org/wiki/Test_plan) to those aggregates, and delivers as its output the integrated system ready for [system testing](https://en.wikipedia.org/wiki/System_testing)

1. when do we use smoke testing and sanity testing?

smoke testing is a **shallow and wide** approach whereby all areas of the application without getting into too deep, is tested. Smoke testing is conducted to ensure whether the most crucial functions of a program are working, but not bothering with finer details.

A sanity test is a narrow regression test that focuses on one or a few areas of functionality. Sanity testing is usually **narrow and deep**. Sanity testing is to verify whether requirements are met or not, checking all features breadth-first.

1. what we will do if we dont have a time to test all stories?
2. what we will do if we come across any severity issue before release day?
3. when do we use automation testing?
4. what tester will do in each phase of SDLC?
5. difference between load and performance testing?

**Web load testing is:**

* similar to, but not synonymous with performance testing
* concerned with the volume of traffic your website (or application) can handle
* not intended to break the system
* viewing the system from the user perspective
* associated with black box testing

**Web performance testing is:**

* a superset of load testing
* concerned with speed and efficiency of various components of the web application
* useful with only one user and/or one transaction
* viewing the system from the architecture perspective (behind the server side curtain)
* associated with white box testing

1. when do we use white box testing and block box testing?

|  |  |
| --- | --- |
| **Black-box testing** | **White-box testing** |
| *Definition:* | Software testing method where the internal structure of the system is**not known** | Software testing method where the internal structure of the system is**known** |
| *Used for:* | Verifying input methods and outputs of the system. | Verifying internal structure     of system’s components |

1. different types of non­functional testing types?

* Baseline testing
* [Compliance testing](https://en.wikipedia.org/wiki/Compliance_testing)
* Documentation testing
* [Endurance testing](https://en.wikipedia.org/wiki/Endurance_testing)
* [Load testing](https://en.wikipedia.org/wiki/Load_testing)
* Localization testing and Internationalization testing
* [Performance testing](https://en.wikipedia.org/wiki/Software_performance_testing)
* [Recovery testing](https://en.wikipedia.org/wiki/Recovery_testing)
* Resilience testing
* [Security testing](https://en.wikipedia.org/wiki/Security_testing)
* [Scalability testing](https://en.wikipedia.org/wiki/Scalability_testing)
* [Stress testing](https://en.wikipedia.org/wiki/Stress_testing)
* [Usability testing](https://en.wikipedia.org/wiki/Usability_testing)
* [Volume testing](https://en.wikipedia.org/wiki/Volume_testing)

1. what is test case?

A **test case**, in [software engineering](https://en.wikipedia.org/wiki/Software_engineering), is a set of conditions under which a tester will determine whether an [application](https://en.wikipedia.org/wiki/Software_application), [software system](https://en.wikipedia.org/wiki/Software_system) or one of its features is working as it was originally established for it to do.

1. what is test plan/test strategy document

|  |  |
| --- | --- |
| **Test Plan** | **Test Strategy** |
| * A test plan for software project can be defined as a document that defines the scope, objective, approach and emphasis on a software testing effort | * Test strategy is a set of guidelines that explains test design and determines how testing needs to be done |
| * Components of Test plan include- Test plan id, features to be tested, test techniques, testing tasks, features pass or fail criteria, test deliverables, responsibilities, and schedule, etc. | * Components of Test strategy includes- objectives and scope, documentation formats, test processes, team reporting structure, client communication strategy, etc. |
| * Test plan is carried out by a testing manager or lead that describes how to test, when to test, who will test and what to test | * A test strategy is carried out by the project manager. It says what type of technique to follow and which module to test |
| * Test plan narrates about the specification | * Test strategy narrates about the general approaches |
| * Test plan can change | * Test strategy cannot be changed |
| * Test planning is done to determine possible issues and dependencies in order to identify the risks. | * It is a long-term plan of action.You can abstract information that is not project specific and put it into test approach |
| * A test plan exists individually | * In smaller project, test strategy is often found as a section of a test plan |
| * It is defined at project level | * It is set at organization level and can be used by multiple projects |

1. what is TDD and BDD (cucumber framework)

TDD- Its also called test-driven design, is a method of software development in which unit testing is repeatedly done on source code. Write your tests watch it fails and then refactor it.**The concept is we write these tests to check if the code we wrote works fine.**

Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design

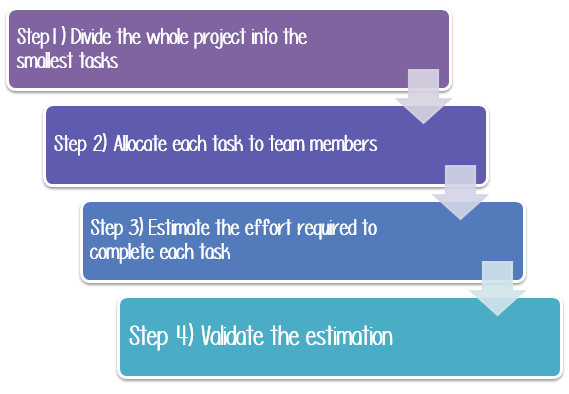
Cucumber lets software development teams describe how software should behave in plain text. The text is written in a [business-readable domain-specific language](http://www.martinfowler.com/bliki/BusinessReadableDSL.html) and serves as documentation, automated tests and development-aid - all rolled into one format.

1. what is priority and severity in defect?

The severity of a bug is a measure of how important the bug is to the end user: how much it breaks, how badly it breaks things, how difficult it is to get work done with this bug in place.

The priority of the bug is a measure of how important the bug is to the development team. This will take into effect the severity, but also the development time and effort, and how much impact fixing it would have on the rest of the product.

1. how to estimate test cases?

[](http://www.guru99.com/images/TestManagement/testmanagement_article_2_2_4.png)

1. what is most challenge defect u came across?

If you came across any such a situation and found any interesting Bug that was difficult to find out or analyzed any project risk accurately before occurring then this could be the answer to this question.

Keep in mind that when answering such a question, be realistic and don’t overstress the situation.

1. what are weakness and strong points

Sir, I am a good team player as well as team leader. I am flexible with situation and cooperative in nature and dedicated towards my work. My strengths are I am a hard worker, good listener.  
  
As every human being have some weakness and my weakness is I want to be perfect. But no body is perfect in this world. But this helps me to improve myself.

1. how to deal the production defects

Handling defects is a pain. The pain increases the more we try to formalise a process to list them, qualify them, order them and ultimately plan them. A complex defect-handling process like this is plain waste. Rather than adding order to the chaos, it actually avoids facing and solving the problems that cause defects.

Instead, fix anything you can within the current Sprint, whether it's a defect or just a better idea - remember to share it with the Product Owner too.

If a defect or a change comes to mind during the Sprint, but without enough time to change it, simply create a Backlog Item for the change and move on. It's the Product Owner's decision whether to accept the item or not, but either way, the upcoming Backlog Item will make it better.

1. test design review steps

**How to review? –**The following are the list of activities involved:

1. Define the criteria – Have a checklist of what to look for?
2. Perform the check
3. Record your results
4. Share, discuss and implement the changes required
5. Version control the documents involved
6. Sign off and use the doc as intended.