**Manual Testing**

1. What is manual testing?

It is used for checking whether the application or the product is working properly or not. We test the application for eliminating the faults or bugs.

1. What types of applications we test?
2. Web based application
3. Windows GUI based application
4. Mobile Application
5. What is SDLC? Diff phases in SDLC?

Software Development Life Cycle- It defines the steps/stsges/phases in building of software.

1. Initial or Requirement
2. Analysis Phase
3. Design Phase
4. Coding Phase
5. Testing Phase
6. Delivery/Maintenance
7. What is waterfall method?

Waterfall model is a popular version of s/w development life cycle model for s/w engineering often. It describes a development method that is linear and sequential.

Once a phase of development is completed the development proceeds to next phase and there is no turning back.

1. What is Agile method?

Agile s/w development is a methodology is a process of developing s/w. Agile means quickly and easily& responding swiftly to change.

Agile projects have sprint or iterations which can shorter in duration (sprint/iteration can vary from 2 weeks to 2 months). Agile projects can have 1 or more iterations and deliver the complete product at the end if the final iterations.

1. Process in agile model?

* Each project is broken into several iterations we can make as many changes as we want .
* All iterations should be of same time duration.
* At the end of each the working product should be delivered.
* In simple terms agile approach the project will be broken into 10 releases.
* Rather than spending 1.5 months on requirements gathering in Agile s/w development, the team will decide the basic core features that are required in product and decide which of these features can be developed in first iteration.
* Remaining features are not delivered in 1st iteration will be taken up by next iteration or subsequent iterations based on the priority.
* In agile we have lot of quick meetings and iterations.
* It is time fixed here the time is allotted simultaneously along with the features.
* Its finished within 2-3 weeks with the final iteration done.

1. What is scrum methodology?

Scrum is an iterative and incremental agile software development framework for managing product development. Scrum is an Agile framework for completing complex projects. Scrum was originally framed for s/w development, but it works for any complex, innovative scope of work. The possibilities are endless.

1. What is daily stand up meeting ?

Daily stand up meeting is short organizational meeting that lasts for about 5 to 15mins each day also called daily scrum or morning roll-call.

Purpose of meeting is:

1. What have you done yesterday?
2. What will you do today?
3. Are there any defects in your work?
4. What is product backlog item?

Product backlog item is a unit of work small enough to be completed by a team in one sprint iteration. Backlog items are decomposed into one or more tasks.

1. What is a user story/feature/sprint backlog items and tasks in user story?

User Story: Is a way to define a s/w feature from an end-user perspective. Example, I want to add a feature to profile page. This gives developers a clear idea of what needs to be developed. User stories are generally added to product backlog/sprint backlog.

Task: Sometimes it makes sense to break a user story down into the work that needs to be done.

1. When do we use automation testing?

Manual testing is costly and time consuming. We use automation tools to reduce cost and repeat no of times with no additional cost.

1. What is sprint planning meeting?

Sprint planning is a collaborating effort involving a scrum master, who facilitate meeting a product owner, who clarifies the details of product backlog items and their respective acceptance criteria and entire agile team who define work and effort necessary to meet their sprint commitment.

1. What is sprint review meeting?

In scrum each sprint is required to deliver a potentially shippable product increment. This means that at end of each sprint the team has produced a coded, tested and usable piece of s/w so that end sprint, a sprint review meeting is held.

1. What is sprint retrospective?

Sprint retrospective is a meeting facilitated by scrum master at which the team discusses the just concluded sprint and determines what could be changed that might make next sprint more productive.

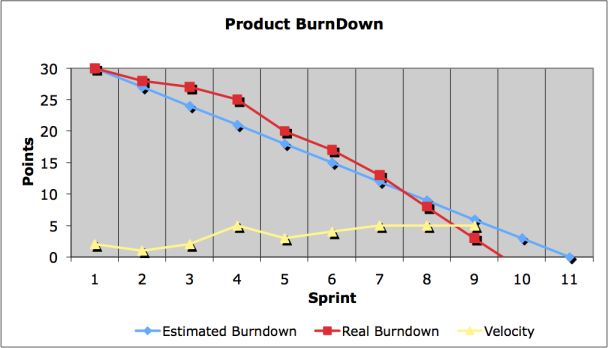
1. What is sprint grooming?

Product **backlog** refinement—sometimes called product **backlog grooming** in reference to keeping the **backlog** clean and orderly—is a meeting that is held near the end of one **sprint** to ensure the **backlog** is ready for the next **sprint.**

1. What is burndown chart and velocity?

The Scrum Burndown Chart is a visual measurement tool that shows the completed work per day against the projected rate of completion for the current project release. Its purpose is to enable that the project is on the track to deliver the expected solution within the desired schedule.

Velocity: The rate of progress of a Scrum Team is called "**velocity**". It expresses the amount of e.g. story points completed per iteration. An import rule for calculating the **velocity** is that only stories that are completed at the end of the iteration are counted.



1. What is UAT? What is user acceptance criteria test cases?

User acceptance is a type of testing performed by the Client to certify the system with respect to the requirements that was agreed upon. This testing happens in the final phase of testing before moving the software application to Market or Production environment.

The main purpose of this testing is to validate the end to end business flow. Identify the test scenarios with respect to high level business process and create test cases with clear test steps. Test Cases should sufficiently cover most of the UAT scenarios. Business Use cases are input for creating the test cases.

1. What is V-Model?

The **V**-**model** is an SDLC **model** where execution of processes happens in a sequential manner in a **V**-shape. It is also known as Verification and Validation **model**. The **V**-**Model** is an extension of the waterfall **model** and is based on the association of a testing phase for each corresponding development stage.

1. What is STLC?

Software Testing Life Cycle (STLC) is defined as a sequence of activities conducted to perform Software Testing.

There is different stages in STLC:



1. What is a 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 erase a defect and what we specify while logging defect?

Defects are removed by the development team as we find the defects while testing we send the defect report to the development team. Defect remove efficiency (DRE) gives the measurement of the development team ability to remove defects prior to the release.

In excel just mention the below things:

Issue ID: Maintain a unique ID for the project.

Summary: What is the bug or issue describe briefly.

Description: explain where the issue has occurred and the situation where it arose with steps to navigate to that issue.

(Description should have steps, Test data, Expected Result and Actual Result)

Screenshots: If u have any attach in the mail.

Priority: Mention the priority based on what you are following in the company like high, medium, low.

Type of issues: like Functional, GUI, DB… etc

Build/Version: Need to mention the build or version ID of the release.

Then send a mail to the concerned person by attaching this excel sheet.

1. What is defect life cycle?

Defect Life Cycle is also called Bug Life Cycle is the journey of defect cycle which a defect goes through during its lifetime. It varies from organization to organization and project to project as it is governed by the s/w testing team and also depends upon the tools used.

1. What is unit testing?

**Unit testing** is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. **Unit testing** can be done manually but is often automated.

1. When do we use regression testing?

**Regression testing** is the process of **testing** changes to computer programs to make sure that the older programming still works with the new changes. **Regression testing** is a normal part of the program development process and, in larger companies, is **done** by code **testing** specialists.

1. What is integration testing?

**Integration testing**, also known as **integration** and **testing** (I&T), is a software development process which program units are combined and **tested** as groups in multiple ways. In this context, a unit is defined as the smallest testable part of an application.

1. Why do we use integration testing?

Most importantly it requires Developer with Tester’s attitude. Management encouragement to build strong Test team who would work and think like a tester with development skill sets. Test Automation Developers need to think beyond writing test after the product built but at very early stage of development. Test are written from the day one of the development cycle.

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

Smoke testing is conducted to ensure whether the most crucial functions of a program are working, but not bothering with finer details. (Such as build verification).

A Sanity test is used to determine a small section of the application is still working after a minor change. Sanity testing is to verify whether requirements are met or not, checking all features breadth-first.

1. What is alpha and beta testing?

This is a form of internal acceptance testing performed mainly by in-house software QA and testing teams. Alpha testing is the last testing done by test teams at development site after the acceptance testing and before releasing the software for beta test. Alpha testing can also be done by potential users or customers of the application. But still this is a form of in-house acceptance testing.

This is a testing stage followed by internal full alpha test cycle. This is the final testing phase where **companies** release the software for few external user groups outside the company test teams or employees. This initial software version is called as beta version. Most companies gather user feedback in this release.

1. When do we use black box testing and white box testing?

Black box testing focuses on the behavior of the software and involves testing from an external or end-user perspective. With black box testing, the tester is testing the functionality of the software without looking at the code or having any knowledge of the application’s internal flows. The term “black box” is used because in this type of testing, you don’t look inside of the application. For this reason, non-technical people often conduct black box testing. Types of black box testing include functional testing, system testing, usability testing, and regression testing.

White box testing validates the internal structure and therefore often focuses primarily on improving security, and making the flow of inputs/outputs more efficient and optimized. In white box testing, the tester is often testing for internal security holes and broken or poorly structured coding paths. The term “white box” is used because in this type of testing, you have visibility into the internal workings. Because of this, white box testing usually requires a more technical person. Types of white box testing include unit testing and integration testing.

1. What we will do if we don’t have time to test all stories/test cases?

**1) Estimate accurately.** When in doubt over-estimate by a reasonable margin, but not underestimate. Don’t forget to make estimate adjustments based on your team, tools and processes. When done, seek official sign off so everyone is aware and is in kept in the loop.

**2)** Take historical data into consideration – **The Test Management tool is your best friend**.

* How long did the earlier release test cycles take?
* What kind of issues caused interruptions to the previous test cycle?
* How many runs did most test cases take before they passed?
* What defects were reported?
* What defects caused the testing to be interrupted?

**3) Ask these questions and plan accordingly in crunch time:**

* Find out Important functionality in your project?
* Find out High-risk module of the project?
* Which functionality is most visible to the user?
* Which functionality has the largest safety impact?
* Which functionality has the largest financial impact on users?
* Which aspects of the application are most important to the customer?
* Which parts of the code are most complex, and thus most subject to errors?
* Which parts of the application were developed in rush or panic mode?
* What do the developers think are the highest-risk aspects of the application?
* What kinds of problems would cause the worst publicity?
* What kinds of problems would cause the most customer service complaints?
* What kinds of tests could easily cover multiple functionalities?

Considering these points, you can greatly reduce the risk of project releasing under less time constraint.

**4) Use a Test Management tool.**This will significantly reduce the amount of preparation, reporting and maintenance time and effort.

1. What we will do when we come across the critical severity issue before the release date?

“severity” is associated with standards. “Severity” is the state or quality of  
being severe; severe implies adherence to rigorous standards or high principles and often suggests harshness; severe is marked by or requires strict adherence to  
rigorous standards or high principles, e.g. a severe code of behavior. The words  
priority and severity do come up in bug tracking. A variety of commercial, problem tracking/management software tools are available. These tools, with the detailed input of software test engineers, give the team complete information so developers can understand the bug, get an idea of its ‘severity’, reproduce it and fix it. The fixes are based on project ‘priorities’ and ‘severity’ of bugs. The ‘severity’ of a problem is defined in accordance to the customer’s risk assessment and recorded in their selected tracking tool.

1. When do we use automation testing?

(Normally, when someone asks this question, we tend to think about automation functional testing tools, like WinRunner, LoadRunner, QTP (Quick Test Pro), Rational Robot, Experian and so on. But the reality is, even a Manual Tester also uses automation tools like bug tracking tools like TestDirector, JIRA, PVC Tracker and so on. Manually repeating these **tests** is costly and time consuming. Once created, **automated tests** can be run over and over again at no additional cost and they are much faster than manual **tests**. **Automated** software **testing** can reduce the time to run repetitive **tests** from days to hours.

1. What tester will do in each phase of SDLC?

1.Tester prepares the Test cases, Test Scenarios from the SRS  
2.  Using the script the tester performs various kinds of testing (Regression, Function)  
3. Tester Notes the results(pass/Fail)  
4. If Result=Fail then the scenario is raised in the Test director   
5. Once its fixed by the developer the tester performs a regression testing.

**34.** What tester will do in each phase of SDLC?

1. Tester prepares the Test cases, Test Scenarios  from the SRS  
2.  Using the script the tester performs different kinds of testing (Regression, Function)  
3. Tester Notes the results(pass/Fail)  
4. If Result=Fail then the scenario is raised in the Test director   
5. Once its fixed by the developer the tester performs a regression testing.

**35.** Difference between load and performance testing?

**Load testing** is the process of putting demand on a software system or computing device and measuring its response. **Load testing** is performed to determine a system's behavior under both normal and anticipated peak **load** conditions.

Performance testing is done to see how long a particular operation takes. How long does it take to request a web page or write some data to the database? It's just an operation and a time.

**36.** Different types of non-functional 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)

**36.** What is test case?

A **test case** is a set of conditions or variables under which a tester will determine whether a system under **test** satisfies requirements or works correctly. The process of developing **test cases** can also help find problems in the requirements or design of an application.

**37.** What is test planning/test strategy document?

A **Test Strategy document** is a high-level **document** and normally developed by project manager. This **document** defines “Software **Testing Approach**” to achieve **testing** objectives. The **Test Strategy document** is a static **document** meaning that it is not updated too often.

**38.** What is Exit and Entry criteria?

**Entry Criteria** for QA testing is defined as “Specific conditions or on-going activities that must be present before a process can begin”. In the Systems Development Life Cycle, it also specifies which **entry criteria** are required at each phase.

**Exit Criteria** is defined as “The specific conditions or on-going activities that must be present before a life cycle phase can be considered complete. The life cycle specifies which **exit criteria** are required at each phase”.

**39.** What is TDD and BDD (cucumber Framework)?

**Cucumber** is a [software](https://en.wikipedia.org/wiki/Software) tool used by [computer programmers](https://en.wikipedia.org/wiki/Computer_programmers) for testing other software. It runs automated [acceptance tests](https://en.wikipedia.org/wiki/Acceptance_testing) written in a [behavior-driven development](https://en.wikipedia.org/wiki/Behavior-driven_development) (BDD) style. Central to the Cucumber BDD approach is its plain language parser called [Gherkin](https://en.wikipedia.org/wiki/Gherkin_(software)). It allows expected software behaviors to be specified in a logical language that customers can understand. As such, Cucumber allows the execution of feature documentation written in business-facing text.

In [software engineering](https://en.wikipedia.org/wiki/Software_engineering), **behavior-driven development** (**BDD**) is a [software development process](https://en.wikipedia.org/wiki/Software_development_process) that emerged from [test-driven development](https://en.wikipedia.org/wiki/Test-driven_development) (TDD).Behavior-driven development combines the general techniques and principles of TDD with ideas from [domain-driven design](https://en.wikipedia.org/wiki/Domain-driven_design) and [object oriented analysis and design](https://en.wikipedia.org/wiki/Object-oriented_analysis_and_design) to provide software development and management teams with shared tools and a shared process to collaborate on software development.

**40.** How do we write test cases in BDD format?

In its simplest format, BDD uses three main descriptors to write a Scenario: Given, When, Then. The text that follows after the descriptors is explained below:

**Given** [context]  
**When** [event]  
**Then** [outcome]

In its simplest format, BDD uses three main descriptors to write a Scenario: Given, When, Then. The text that follows after the descriptors is explained below:

**Given** [context]  
**When** [event]  
**Then** [outcome]

 ‘Ands’ to explain all the events that are required to be met before an account can be created.

**Given** Fred is signing up for Facebook,  
**When** he enters his First Name,  
**And** enters his Last Name,  
**And** enters his Email,  
**And** re-enters his Email,  
**And** enters a new Password,  
**And** specifies his Gender,  
**And** enters his Birthday,  
**And** submits his request,  
**Then** a Facebook account is created,  
**And** account name is set as Fred’s email address.  
**And** a confirmation email is sent to Fred.

**41.** what is priority and severity in defect?

Usually the severity is defined in terms of financial loss, damage to environment, company's reputation and loss of life. **Priority** of a **defect** is related to how quickly a bug should be fixed and deployed to live servers. When a **defect** is of high severity, most likely it will also have a high **priority.**

**Severity** is defined as the degree of impact a **defect** has on the development or operation of a component application being tested. Higher effect on the system functionality will lead to the assignment of higher **severity** to the **bug**.

**42.** How to estimate test cases?

CA Agile Central's approach is to create a [task](https://help.rallydev.com/glossary#task) under the [use story](https://help.rallydev.com/glossary#user_story) for testing tasks and associate the test cases to the user stories.

Test Case Estimate

* + Display Name: TestCaseEstimate
  + Type: Decimal

Test Case To Do

* + Display Name: TestCaseToDo
  + Type: Decimal

Test Case Actual

* + Display Name: TestCaseActual
  + Type: Decimal

**43.** What are test design techniques?

By design we mean to create a plan for how to implement an idea and technique is a method or way for performing a task. **Test Design** is creating a set of inputs for given software that will provide a set of expected outputs. The idea is to ensure that the system is working good enough and it can be released with as few problems as possible for the average user.

**44.** If we don’t have time to test call test cases what we will do?

1. Have management define priorities.
2. Look for duplicate coverage. Remove redundant tests.
3. Use test cases with the most coverage.
4. Enlist help from other teams if you can. E.g. have the documentation team walk through their documentation steps.

**45.** How we learn the functionality of the system?

**Functionality testing** is performed to verify that a software application performs and functions correctly according to design specifications. During functionality testing we check the core application functions, text input, menu functions and installation and setup on localized machines, etc.

**46.** Who will assign the work?

Test Lead will assign the work to test engineers

**47.** What is the requirement traceability matrix?

The **Requirements Traceability Matrix** (RTM) is a document that links **requirements** throughout the validation process. The purpose of the **Requirements Traceability Matrix** is to ensure that all **requirements** defined for a system are tested in the test protocols.

**48.** What are the typical environments we have on the projects?

**49.** What is QA environment?

A **QA environment** is where you test your upgrade procedure against data, hardware, and software that closely simulate the Production **environment** and where you allow intended users to test the resulting Waveset application. A Production **environment**s where the Waveset application is actually available for business use.

**50.** What is staging environment?

A **stage** or **staging environment** is an **environment** for **testing** that exactly resembles the production **environment**. In other words, it's a complete but independent copy of the production **environment**, including the database. **Staging** provides a true basis for QA **testing** because it precisely reproduces what is in production.

**51.** What is production environment?

**Production environment** is a term used mostly by developers to describe the setting where software and other products are actually put into operation for their intended uses by end users.

**52.** What is production environment?

**Production environment** is a term used mostly by developers to describe the setting where software and other products are actually put into operation for their intended uses by end users.