**MANUAL TESTING NOTES**

**---------------------------------------------------------------------------------------------------------------------**

~Sids

**1)Why testing is required?**

Ans: **Testing is required** for an effective performance of software application or product. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development. It's **required** to stay in the business

**2) What types of application we test**

Ans: web applications,

desktop/windows applications

Mobile applications

ETL jobs

Back end/batch programs/windows services

**3) what is SDLC and different phases in SDLC?**

Ans: Software development life cycle (SDLC) is a process to develop the application

**Different phases like:**

**Requirement Analysis and planning :** Senior team members analyze the requirements/input given by customers/business users. They will check whether the requirement is feasible or not (can be done or not). They also identify the risks associated with project.

Note: this high level requirements will be written in BRD (Business Requirement document) by Business Analyst

Define/Design : in the define stage Business Analyst define more details about requirements (which are in BRD) in the form of SRS (software requirement specification) or Use Case diagram.

As part of design,

Senior Developers write High Level Design Document (HLD)

Developers write Low Level Design Document (LLD)

Seniors Tester write Test Planning document

Implementation/Development: Developers write the code for the requirements

Testers write test cases as per SRS

Testing : Execute the test cases what we prepared in previous stage

Deployment : Release the tested code to production

Maintenance : Support team monitoring the system that is running in production

**2) what is waterfal in SDLC?**

Ans: The **waterfall** model is a sequential (non-iterative) design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a **waterfall**) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

**what is the process in agile model**

**Ans: Agile development model** is also a type of [**Incremental model**](http://istqbexamcertification.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/). Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly [**tested**](http://istqbexamcertification.com/why-is-testing-necessary/) to ensure [**software quality**](http://istqbexamcertification.com/what-is-software-quality/) is maintained. It is used for time critical applications.  Extreme Programming (XP) is currently one of the most well known agile [**development life cycle model**](http://istqbexamcertification.com/what-are-the-software-development-models/).

**what is scrum methodology**

Ans: Scrum is an **agile** software development model based on multiple small teams working in an intensive and interdependent manner. The term is named for the scrum (or scrummage) formation in rugby, which is used to restart the game after an event that causes play to stop, such as an infringement.

**what is daily standup meeting and what we discuss**

Ans: 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.

what is user story/feature/sprint back log items and tasks in user story

what is sprint planning and spring retro

**Sprint planning** is a collaborative effort involving a Scrum Master, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who **define** the work and effort necessary to meet their **sprint** commitment

what is burndown chart and velocity

what is product backlog item and sprint backlog items

what is user acceptance criteria test cases

Firstly, the **criteria** by which the software is considered to be “working” needs to be assembled. These are likely to be collated from the system requirements, and **user**stories. Next, a set of **UAT test cases** must be created. Centric defines a **UAT test case** as: ... Each **case** covers a specific usage scenario of the software.

**what is v model?**

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

**what is STLC?**

**Software Testing Life Cycle** (**STLC**) is the testing process which is executed in systematic and planned manner. In **STLC** process, different activities are carried out to improve the quality of the product. Let's quickly see what all stages are involved in typical**Software Testing Life Cycle** (**STLC**)

**what is defect?**

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**. When actual result deviates from the expected result while **testing** a**software** application or product then it results into a **defect**.

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

**Defect logging**, a process of finding defects in the application under test or product by testing or recording feedback from customers and making new versions of the product that fix the defects or the clients feedback.

**Defect tracking** is an important process in software engineering as Complex and business critical systems have hundreds of defects. One of the challenging factors is Managing, evaluating and prioritizing these defects. The number of defects gets multiplied over a period of time and to effectively manage them, defect tracking system is used to make the job easier.

**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

**Different types of testing:**

**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** is often automated but it can also be done manually.

**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.

What is integration testing?

**Integration testing** (sometimes called **integration** and **testing**, abbreviated I&T) is the phase in software **testing** in which individual software modules are combined and **tested** as a group. It occurs after unit **testing** and before validation **testing**.

when do we use integration testing?

when do we use smoke testing and sanity testing?

**Smoke Testing**, also known as “Build Verification**Testing**”, is a type of software **testing** that comprises of a non-exhaustive set of **tests** that aim at ensuring that the most important functions work. The results of this **testing** is used to decide if a build is stable enough to proceed with further **testing**.

**Sanity Testing** is the subset of Regression **Testing** and it is performed when we do not have enough time for doing **testing**. **Sanity testing** is the surface level **testing**where QA engineer verifies that all the menus, functions, commands available in the product and project are working fine

what is UAT?

In software development, user acceptance testing (UAT) - also called **beta** testing,**application** testing, and end user testing - is a phase of software development in which the software is tested in the "real world" by the intended audience.

what is alpha and beta testing?

**Alpha testing** is a type of acceptance **testing**; performed to identify all possible issues/bugs before releasing the product to everyday users or public. ... **Alpha testing** is carried out in a lab environment and usually the testers are internal employees of the organization.

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

Black-box testing (also known as functional testing) treats software under test as a black-box without knowing its internals. Tests are using software interfaces and trying to ensure that they work as expected. As long as functionality of interfaces remains unchanged, tests should pass even if internals are changed. Tester is aware of what the program should do but does not have the knowledge of how it does it. Black-box testing is most commonly used type of testing in traditional organizations that have testers as a separate department, especially when they are not proficient in coding and have difficulties to understand the code. It provides **external perspective** of the software under test.

what we will do if we don’t have a time to test all stories?

what we will do if come across any severity issue before release day?

You must -absolutely- not ever hold off from reporting an issue, at least to your local management structure. That would, at the very least, ruin the reputation of your team and could potentially have much more serious consequences.

The next thing to do is to determine the answer to the obvious question: "Why was this found so late?". There are many reasons why this situation could arise - your test preparation could've been too light, you could've mis-prioritised some work, there may simply have been too much to do. As a member of the test team, you need to know what caused the issue and therefore how you can reduce the risk of it happening again.

Obviously, we don't live in a perfect world and it's possible that no action may be taken to resolve the defect before release. It's entirely possible that it makes more sense to go live with a bug and then release a quick fix, than to decide not to release at all. That's a different discussion, though.

For what it's worth, this has happened to me a couple of times over the past dozen or so years. It's a horrible feeling but as long as you've done your best, there's nothing more you can do.

when do we use automation testing?

Test engineers strive to catch them before the product is released but they always creep in and they often reappear, even with the best manual **testing processes**. Test Automation software is the best way to increase the effectiveness, efficiency and coverage of your software testing.

what tester will do in each phase of SDLC?

Phase will start from Requirement stage itself

1. Requirement stage - PM,Tech.Lead, Testlead will review the document

2. Design Stage - PM or Test lease will review the document

3. coding Stage - Testlead will prepare Testplan, Tester will prepare testcases

4. Testing Stage - Tester will execute the testcase

difference between load and performance testing?

Performance testing is the testing, which is performed, to ascertain how the components of a system are performing, given a particular situation. Resource usage, scalability and reliability of the product are also validated under this testing. This testing is the subset of performance engineering, which is focused on addressing performance issues in the design and architecture of software product.

Load testing is meant to test the system by constantly and steadily increasing the load on the system till the time it reaches the threshold limit. It is the simplest form of testing which employs the use of automation tools such as LoadRunner or any other good tools, which are available. Load testing is also famous by the names like **volume testing** and **endurance testing**.

different types of non-functional testing types?

* Load/Performance testing.
* Compatibility testing.
* Localization testing.
* Security testing.
* Reliability testing.
* Stress testing.
* Usability testing.
* Compliance testing.

what is test case?

A test case is a document, which has a set of test data, preconditions, expected results and postconditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

**what is test plan/test strategy document**

Ans: Test plan document contains different section like

Types of testing :

Exit and Entry criteria :

**Test strategy** is a high level **document** which defines the approach for software**testing**. It is basically derived from the Business Requirement **document**. **Test strategy** is developed by project manager or business analyst. It is kind of static**document** which sets the standards for **testing** so not updated often.

what is TDD and BDD (cucumber framework)

BDD:

In software engineering, **behavior-driven development** (abbreviated **BDD**) is a software development process based on test-driven development (TDD). Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design to provide software development and management teams with shared tools and a shared process to collaborate on software development.

TDD:

**Test-driven development** (**TDD**) is a software development process that relies on the repetition of a very short development cycle: first the developer writes an (initially failing) automated test case that defines a desired improvement or new function, then produces the minimum amount of code to pass that test, and finally refactors the new code to acceptable standards. Kent Beck, who is credited with having developed or ‘rediscovered’ the technique, stated in 2003 that TDD encourages simple designs and inspires confidence

what is priority and severity in defect?

**Severity** is defined as the degree of impact a **defect** has on the development or operation of a component application being tested.

how to estimate test cases?

1. 3-Point Software Testing Estimation Technique.
2. Use – Case Point Method:
3. Work Breakdown Structure.
4. Wideband Delphi technique.
5. Function Point/Testing Point Analysis.
6. Percentage of development effort method.
7. Percentage distribution.
8. Best Guess.

what is most challenge defect u came across?

**how to deal the production defects?**

Ans: normally end user will report this issue.

we need to talk to them (end users) and reproduce the issue with in staging environment

Create defect in defect tool under the production release version

developers will fix the issue

we (QA) test the issue on production version code (stageing) and release the fix to proudction after we verify

we have to create a defect on current **sprint/release** so that developer will add this code to the current sprint/release

test design techniques:

So, **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. Broadly speaking there are two main categories of **Test Design Techniques**.

**If we dont have time to test call test cases what we will do**

**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 is 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.

=> **For the list of the most popular test management tool choice**, [check out here](http://www.softwaretestinghelp.com/15-best-test-management-tools-for-software-testers/):

**#5)** There is not much we can do about incorrect builds/technical issues, but the one thing that can help is looking at the Unit test results. This will give us an idea as to whether the build was a success or not and what kind of tests did it fail – so we don’t reinvent the wheel.

If your **Test Management Tool supports**[**CI integration**](http://www.softwaretestinghelp.com/continuous-integration/), you have that information available without any fuss so you understand the stability of the application better.

**#6) Measure your productivity and progress often**. Don’t let status reports be a deliverable just for the benefit of the external teams. Make sure you are closely monitoring your daily targets and your ability to accomplish them.

Also, be sure to not get into the classic conundrum of ‘Velocity vs. Quality’. Because, when you report, say, 50 bugs a day, it might appear as if you are being super productive. But if most of them are coming back as invalid ones, you have got yourself a problem.

how we learn the functionality of system?

what are the tools to manage defects/stories?

|  |  |
| --- | --- |
| **Product** | **Vendor** |
| Bugzilla | Bugzilla.org |
| Census BugTrack | MetaQuest |
| Lean Testing | Crowdsourced Testing |
| DefectTracker | Pragmatic Software |

45 more rows

who will assign the work?

Generally test lead assign the work to test engineer. It   
depends on the organisation where Test lead and Test   
enginerrs combindly may start the process based on SRS and   
DDD

types of test metrics we use normally

1. Base Metrics
2. Calculated Metrics

**Base Metrics:**  
Base Metrics are the Metrics which are derived from the data gathered by the Test Analyst during the test case development and execution.

This data will be tracked throughout the Test Life cycle. I.e. collecting the data like, Total no. of test cases developed for a project (or) no. of test cases need to be executed (or) no. of test cases passed/failed/blocked etc.

**Calculated Metrics:**  
Calculated Metrics are derived from the data gathered in Base Metrics. These Metrics are generally tracked by the test lead/manager for Test Reporting purpose.

what is traceability matrix?

A **traceability matrix** is a document, usually in the form of a table, used to assist in determining the completeness of a relationship by correlating any two baselined documents using a many-to-many relationship comparison.

what are typical environments we have in projects

Identify activities

• Estimate times and resources

• Identify relationships and dependencies

• Identify schedule restraints

• Create the schedule

what are different defect metrics and measurements we prepare

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.

what is development environment

In computer program and software product **development**, the **development environment** is the set of processes and programming tools used to create the program or software product. The term may sometimes also imply the physical**environment**

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**is where the Waveset application is actually available for business use.

what is production environment

A **production environment** is where the real-time staging of programs that run an organization are executed, and includes the personnel, processes, data, hardware, and software needed to perform day-to-day operations.