### **manual testing-**

Before releasing software to the public, it is vital to test it manually. The key subjects to cover in this regard are sdlc and style.

**SDLC stands for software development life cycle**. There are various life cycles, such as agile, spiral, waterfall, v, and prototype, that are used to build and deliver software.

i- waterfall: It is a fundamental model that resembles our college academic projects; the model’s flow is as follows:

The customer/client will speak with a business analyst, who will then create a CSR (customer requirement specification that includes low-level diagrams) and discuss it with management (HR, Team study, and necessities). Design engineers will then convert the low-level diagrams into high-level diagrams (software requirement specification) and give them to the coding team, which will then give to the testing team. Finally, the built (software that is prepared for testing) will be given to the installation team, who will then install it.

(if the client had any issues he can reach out to the maintenance engineer until the contract ends)

ii- prototype: In this stage of testing, the customer or client’s requirements are discussed, and the prototype (a dummy version of any software referred to as a prototype) is released into their hands. If it meets their needs, it moves on to the coding, testing, installation, and maintenance stages; otherwise, the process is updated to include the customer’s most recent requirements in the prototype and design. This cycle is repeated until the customer or client is satisfied.

iii. spiral: if any phase wants to alter, only the required component is added, and the cycle is repeated until the client approves, the build is divided into several pieces and follows requirement — -design — code — test.

Iv. The verification and validation model, or v model, are additional names that can be used to describe this.

This would go through several phases.

1- crs — srs (copy delivered to test engineer to do acceptance testing)

2. srs — hld (copy provided to test engineer for system testing)

3. hld — lld (copy provided to test engineer to conduct integration testing)

4. lld — code (copy sent to test engineer to perform functional testing)

5. coding — code (copy handed to test engineer to perform unit testing)

Coding leads to building a product that is released to the market.

V. agile model

The goal here is necessity Each phase of development, coding, and design is tested until it satisfies the client’s requirements.

SCRUM METHODOLOGY

the framework that is used to build the software’s

this has

1. product owner: accept /reject the project
2. scrum master: use to check whether an agile process is going on well/not
3. developers: design and execute the codes
4. testers: write test cases and meet the requirements

the common terms used in this are :

1. user story: a collection of requirements
2. epic: the collection is stories
3. product backlog: a collection of user stories prepared by the product owner
4. sprint: time taken by the product owner & scrum master to assign tasks to the team
5. sprint meeting: in this meeting what work need to be done was discussed
6. sprint backlog: list of committed stories done by developers and testers for a specific sprint
7. scrum meeting: meeting conducted by scrum master for 3 questions

i.what did you do yesterday?

ii. what will you do today?

iii are there any obstacles in the process?

8. burnup/burndown chart: maintained by ScrumMaster daily helps to analyze how much work remains

9. sprint retrospective meeting: meeting conducted after completion of 1 sprint.

process :

Agile project management is facilitated by the Scrum framework, which encourages adaptability, openness, and teamwork. The development process is separated into brief iterations known as sprints, during which the team concentrates on a single functional component. A prioritized list of features that need to be created can be found in the product backlog. The daily stand-up meetings make sure that everyone is on the same page and that any problems are dealt with right away. Although Scrum is frequently used in software development, it can also be employed in other fields.

on simple terms

SDLC follows

1. requirement analysis
2. define
3. design
4. implementation
5. testing
6. deployment
7. maintenance

The other important term in manual testing is **software testing life cycle**

[standard test case template — Google Sheets](https://docs.google.com/spreadsheets/d/1w5sGSysrZZjDCQtUF3FmVGXh6o2iVypXB0Z-54FIaE0/edit#gid=0)

1. requirement study: understand the requirement to perform any testing

Use Case Diagram: here it will specify what functionalities are performed by what actors

2. Test plan: document written to perform testing in an effective, efficient manner

3. test case:

1. requirement
2. identify all possible scenario
3. write test cases by using test case design
4. review test case
5. test case approval
6. store test case repository

Design Techniques to do testing

I .boundary value analysis: consider boundary values to perform testing

here we will select 1 less than the minimum and 1 more than the maximum.

ii. equivalent class partitioning: here test cases are divided into a finite number of data classes this testing is performed to reduce the infinite number of cases into a finite number of cases

iii. error guessing technique: here we need to think about a situation where how software behaves in the future(normally it is given to experienced persons)

here senior testing engineer will send them to his companion/friend who is another senior testing engineer in the same organization the other person will review the document(which contains 2 columns test case id and comment in the comment he will comment checking whether his colleague had done testing according to a requirement or not)

then senior shares in shared lock folder to the junior test engineer

test execution report: here junior test engineer fills the columns of status, and defect number severity.

4. Traceability matrix: a document that maps requirement number and test case id

depending upon areas of mapping divided into

1. requirement traceability matrix: requirement num and test case id used to check every requirement associated with it or not.
2. defect traceability matrix: defect num and test case id to re-execute the test cases defects are fixed.

5. defect tracking: the process of isolating and managing the defects when they are they need to be sent to a required developer by following ways.

1. writing a mail
2. using tool like (jira,mantis…)

6. test execution report: it shows the status of executed test cases maintained daily

other types of testing

black box testing: testing done without knowing the code

other names: non-code, nontransparent, nonstructural, closed box

functional: software is tested against the functional requirement

user acceptance: user will test the functionality of the software in presence of a client

here is alpha: testing done at the developer side office & beta: testing done at the product side office

system testing: do testing in the client environment and perform testing to it, here complete business functionality would be tested

integration testing: a combination of 2 or more modules to check data flow is happening is called integration testing

white box testing: testing done by knowing the code

statement testing: here we need to check whether every statement is executed at least once.

path testing: testing is done by checking the flow diagram.

loop testing: test engineer will pass testing on the looping control construct

here we will check

1. the loop should be executed exactly more than once.
2. the loop should be executed at least once.
3. the loop should fail once,

conditional testing: here test engineer will execute the testing on or group of tests on a simple /complex conditional statement and check whether every truth value is getting executed at least once/not.

other names: open box, code-based, structurally based, transparent

Gray box testing: the combination of white and black box testing techniques

other names: white +black

other important testing definitions

smoke testing: testing done before accepting the build

sanity testing: testing is done after accepting the build

Regression testing: testing again and again on different builds

here we have

unit regression testing: performing testing only on the affected module

regional regression testing: performing testing on the affected module along with the relational module

full regression testing: performance testing on the entire build

(since it requires partial code it belongs to gray box testing)

compatibility: test eng will check whether the test built is compactable to different versions

functional testing: testing will be done on the functionality of the software application.

nonfunctional testing: testing can be done on quality aspects of software

retesting: testing performed on the same build again and again by giving similar inputs.

usability testing: here test engineer will check for 2 things

1. how easily software use
2. how friendly software to learn

static testing: test the application-related factors

dynamic testing: performing some actions( all black box and white box techniques)

globalization testing: performing testing on software that allows multiple languages

localization testing: test engineer will check whether it follows the local format (currency, Pincode, phone num, date, and time)

recovery testing: breaks the working system and immediately corrects it

authentication .direct URL and firewall leakage are the 3 types of aspects checking in recovery testing

performance testing: check the response and stability with workflow

1. load: built is tested with various loads
2. soak: the app is tested with the same load and different intervals of time
3. spike: the app is tested with loads at a certain time for measuring performance
4. stress: built is tested beyond the maximum capacity for increasing stress

Bug Detection Cycle: When a defect is discovered, it is assigned to the appropriate developer, who then investigates it, fixes the problem, tests it, and moves on to the confirmed stage, which is followed by the dead state. If the scenario is not met, the bug is reopened and checked to see if it can be reproduced. If it cannot, it is in the rejected state, and if it does not affect the front end, it is in the deffered state.The software tester will continue the procedure of restest state after he receives the progressed bugs.

Unit testing: in this every functionality of the software are tested, the main purpose of unit texting is to validate the software core as performed and we got the expected output unit testing is done by the developers in coding stage.

exploratory testing: it is the type of testing when test engines are not created in advance but the test engineer need to check the testing process of the system. (scenario may be they may have to note down the ideas as what to test before test excution rather than testing it is like a thinking act, used in agile model which is all about investigation,discovery,learning.

ADHOC testing/monkey testing: testing performed on software by breaking to check the stability of the software is called adhoc testing

Test driven development: software is made by developers from the testcases

the scenario will be as if the developer starts wrting the code for the testcase then tester will run the test caseand if fails he asks the developer to make some changes in the code and this cycle will be continues till the completion of all testcases

Here is a basic template that had been done by me kindly check this out and let me know if there are any errors.

[here is the project link : https://bit.ly/3Ap89We](https://bit.ly/3Ap89We)