STLC ⇒ It is sequence of specific activities conducted during testing process to ensure s/w quality goals are met.

STLC identifies what test activities to carry out & when to accomplish those test activities.

Phases in STLC

① Requirement Analysis ⇒

entry criteria ⇒ BRS

→ During this phase, test team studies & analyze the requirements from a testing perspective/pointofview.

→ This phase helps to identify whether the requirements are testable or not.

→ During this phase if any requirement is not testable test team can communicate with various stakeholders such as client, BA, technical leads etc.

② Test planning ⇒ It is first step of the testing process

⇒ In this phase typically Test Manager / Test lead involves to determine the effort and cost estimates for entire project.

⇒ Preperation of test plan will be done based on the requirement analysis.

⇒ Activities like resource planning, determining roles & responsibilities, tool selection (if Automation), training requirement etc. carried out in this phase.

③ Test Design ⇒

⇒ Test team starts with test case development.

⇒ Test team prepares test cases, test scripts (if automation) and test data.

⇒ Once the test cases are ready then these test cases are reviewed by peer members or team lead.

⇒ Also test team prepares the Requirement traceability matrix also known as RTM.

⇒ RTM traces the requirements to the test cases that are needed to verify whether the requirements are fulfilled.

④ Test Environment setup ⇒

⇒ This phase can be started in parallel with test design phase.

⇒ Test environment setup is done based on the h/w & s/w requirement list.

⇒ In some cases, test team may not be involved in this phase.

⇒ development team or customer provides the test environment.

⇒ meanwhile test team should prepare the smoke test cases to check the readiness of the given test environment. & smoke test results

⑤ Test Execution ⇒

⇒ Test team starts executing the test cases based on the planned test cases. if test case result is pass/fail then the same should be prepared for failed test cases & should be reported to dev team through bug tracking tool for fixing the defects. Retesting will be started/performed once the defect was fixed. test case execution report, defect report, RTM.

⑥ ~~Test~~

Test closure ⇒

⇒ this is final stage. here we prepare test closure Report, Test metrics.

⇒ Testing team will be called out for a meeting to evaluate cycle completion criteria based on Test coverage, Quality, Time, Cost, S/W, Business Objectives.

⇒ Test team analyses the test artifacts (such as test cases, defect reports etc) to identify strategies that have to be implemented in future, which will help to remove process bottlenecks in the upcoming projects.

⇒ Test metric & test closure report will be prepared based on the above criteria

2 types of S/W test metrics ⇒

① process metrics ② product metrics

Pro

Tracebility matrix ⇒

Application ⇒ 100 modules ⇒ for each module there is test Scenarios ⇒ for each test scenarios there is test case

we need to have some document to showcase it to client ok these are all the features/modules we have tested

Tracing the requirements whether you have covered testing or not

Req ID ⇒ All req id ins here

tracing requirement whether you have covered the test case writing & scenario writing or not, that you are going to verify by updating the table.

we can able to trace our requirements whether ~~it~~ is all the ~~Scenarios are~~ requirements are covered for scenarios writing or for test case writing or not we will verify by putting it in table format if at all each & every requirement is having corresponding Scenario ID corresponding test case IDs that requirement is covered for scenario & test case writing if any requirement is doesn't have any Scenario ID it means that requirement you get missed in all writing test scenarios and test cases

Requirement ID	Scenario ID	Test Case ID

(RTM) => cross Reference matrix

It is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled.

=>

① forward Traceability => mapping requirements to test cases is called forward ~~page~~.

it is used to ensure whether the project progress is in the desired directn.

it ensures that each test case tested throughly.

② Backward/ Reverse traceability =>

=> mapping test cases to requirement.

=> it is used to ensure whether the current product remains on right track

it make sure we are not expanding scope of the project. by adding functionality that is not specified in requirements.

③ Bidirectional trac =>

=> it is used to ensure that all the specified requirements have appropriate test cases & vice versa.

adv => ① 100% test coverage

② It allows to identify the missing functionality

③ It is easy to track the overall test executn status.

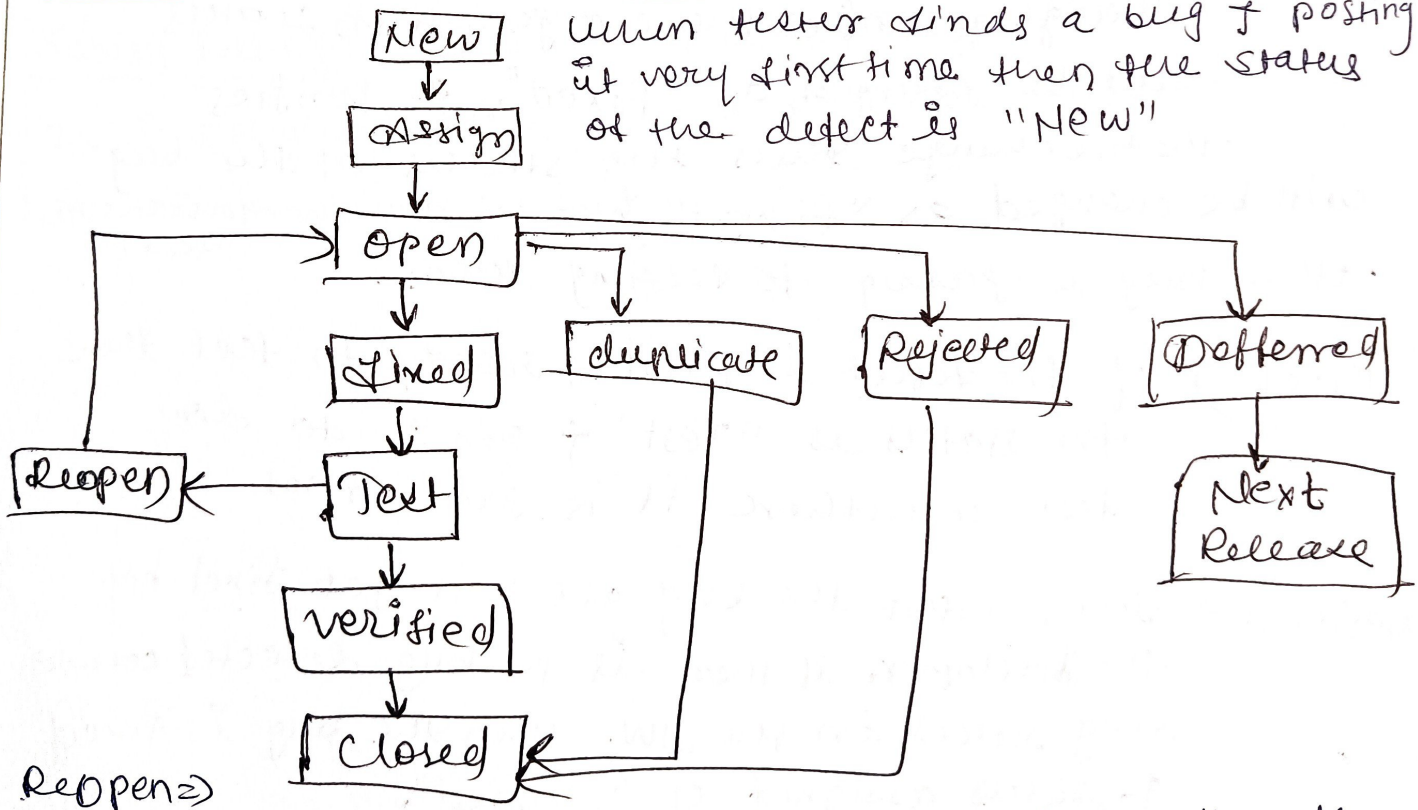
Bug life cycle ⇒ defect life cycle

⇒ defect lifecycle in SW development process the bug has a life cycle.

⇒ The bug should go through the life cycle to be closed.

⇒ Name of the bug status may vary depends upon the tools (QC, JIRA etc) used and the process followed in the organizat^n.

> Bug can be defined as the abnormal behaviour of the SW. Bug starts when defect is found & ends when a defect is closed, after ensuring it is not reproduced.



when tester finds a bug & posting it very first time then the status of the defect is "New"

Reopen ⇒ if the defect Remains same after the retest, then the tester changes the status to "Reopen", again bug go through the life cycle to be fixed.

duplicate ⇒ if the defect is repeated then the status is changed to duplicate by dev. team

Assigned ⇒ defects which are in the status of New will be ~~prod~~ approved if valid & assigned to development team by test lead, project lead or project manager.

Status of bug changes to assigned if the development team accepts the bug then bug states changes to "open".

Open ⇒ once the development team starts analyzing & works on the defect fix then the status is "open"

fixed ⇒ when developer makes necessary code change & verifies the change, then status will be changed as "fixed". & verifies the the change then the status of the bug will be changed as "fixed" here the dev change status to "test" the bug is passig to testing team.

Test ⇒ if the defect is fixed & ready to test then the status is "Test" & ready to do test whether it is fixed or not.

verified ⇒ tester retest the bug after it got fixed by the developer. if there is no bug detected while doing retest in the SIW, then the bug is fixed & status assigned is "verified".

closed ⇒ after verified the fix, if the bug is no longer exists then the status of bug will be assigned as "closed"

Sometime while doing retest we may encounter same issue again which means bugs not ~~fixed~~ fixed properly then tester make changes status as Reopen

Smoke Te
it will
perform o
It mean
of a app

Sanity Te
It is sp
in deta

Assigned ⇒ defects which are in the status of New will be ~~prod~~ approved if valid & assigned to development team by test lead, project lead or project manager.

Status of bug changes to assigned if the development team accepts the bug then bug states changes to "open".

Open ⇒ once the development team starts analyzing & works on the defect fix then the status is "open".

fixed ⇒ when developer makes necessary code change & verifies the change, then status will be changed as "fixed". & verifies the the change then the status of the bug will be changed as "fixed" here the dev change state to "test" the bug is passing to testing team.

Test ⇒ if the defect is fixed & ready to test then the status is "Test" & ready to do test whether it is fixed or not.

verified ⇒ tester retest the bug after it got fixed by the developer. if there is no bug detected while doing retest in the SIW, then the bug is fixed & status assigned is "verified".

closed ⇒ after verified the fix, if the bug is no longer exists then the status of bug will be assigned as "closed"

Sometime while doing retest we may encounter same issue again which means bugs not ~~fou~~ fixed properly then tester make changes states as Reopen

Smoke T

it will
perform
It mean
of a ap

Sanity Te

It is sp
in data

Smoke Testing →

it will done after getting build. it is high level testing perform on application.

It means basic testing or checking main functionality of a application. here we don't do deep dive testing.

Sanity Testing ⇒

It is a subpart of regression testing. It is specialized testing. test a particular functionality in detail.

Retesting ⇒ testing specific bug after it is fix.

To ensure that the defects which were found & posted in the earlier build were fixed or not in the current build.

for eg. tester find bug in build 1.0, posted defect 1.0.1 & defe 1.0.2 ...

⇒ It is planned testing

When? ⇒ ① when there is a particular bug fix specified in the Release Note.

② when a bug is rejected.

③ when a ~~cle~~ client calls for retesting.

Regression Testing?

⇒ Repeated testing of an already tested program, after modificatn

⇒ When ⇒ ① we do regression whenever the production code is modified.

Case 1 ⇒ when new functionality added to the applicatn.

case 2 ⇒ when there is change requirement

case 3 ⇒ when there is defect fix.

case 4 ⇒ when there is performance issue fix.

Loading the Home page takes 5 sec, Reducing the load time to 2 seconds

case 5 ⇒ when there is Environment change.

We do Regression testing after every deployment.

Black box	White box
① It is called a Behavioural / Specificat^n based / I/P-O/P testing	① It is called as Glass Box, Clear box, Structural testing.
② tester performs testing only on the functional part of an applicat^n to make sure the behaviour of s/w is as expected.	② It is based on applications internal code structure. usually done at unit level.
technique ⇒ ① equivalence ② BVA ③ Decision table ④ State transition.	③ techniques ⇒ ① statement coverage ② brach coverage ③ path coverage
2 types ⇒ ① functional ② Non functional	
functional ⇒ what the system actually does is	
Non funct ⇒ How well the system performs is	

Smoke Testing ⇒ It is done to make sure if the build we received from the development team is testable or not. It is also called as "Day 0" check. It is done at "build level".

eg. ⇒ In project for the first release, development team releases the build for testing & the test team tests the build. Testing the build for the very first time is to accept or reject the build. this we call it as Smoke testing.

Sanity testing ⇒ it is done during the release phase to check for the main functionalities of the applicat^n without going deeper. it is also called as subset of Regression testing. It's done at release level. [At the times due to release time constraints rigours regression testing can't be done to the build, sanity testing tests does that part by checking main functionalities]

test main functionality without going deeper is called sanity testing.

what is testing => testing is to deliver a quality product to customer. if ~~verify the applicatn~~ any deviation, we need to identify & report it to development team get it fix & we need to cross check again our intention is to deliver a 100% quality application to the client.

How you will confirm application is meeting all the requirements?

~~at~~ applicatn all the requirement given by the client. we are deriving the test cases if all the tests ~~cases~~ cases are passed & no open bugs are there then we can confirm met all the requirements.

everything working is fine as per requirement.

when you stop your testing?

As per traceability, All the test cases are passed and there is no open issues then we will stop our testing.

→ first of all trace or check traceability matrix whether all the test cases are prepared or not can & every requirement is covered or not. All the test cases executes, some of them will be passed some of them will be failed, then update it to development team & get fix & then perform retesting. then nothing is there to test whatever we test if everything is working fine then we are going to stop testing.

→ based on 100% test coverage is there, then we can stop testing if there are no open bugs

difference betn project and product?
project is for single Customer Requirements and where as product is for market industry requirements.

→

what is smoke testing? when we do it.
~~before~~ developer will developed applicatn & given to testing team. before we are accept the build or sprint to the testing. we are going to perform initial testing on it. atleast, basic functionalities are working or not to accept verify it for major testing ~~then~~ it is first type of testing perform on applicatn whether to verify build is stable or not. accept the build for major testing or not.

⇒ build is verify to accept major testing or not.

diff. betn GUI and functional testing. - footer, header

GUI testing → verify, button, text fields, links, logos radiobuttons, labels, drop down, comboboxes, checkboxes

functional testing ⇒ verify applicatn behaviour. whether it is working according to the customer requirement

Regression testing ⇒ entire applicatn is stable or not.

How will you identify related functionalities in Regression testing?
by using decision table, we will identify what are the functionalities are inter related. when bug is fixed, we will refer decision table & execute the test cases accordingly in the part of regression testing.

What is adhoc testing?
When there is no enough time, we will test functionalities which are imp (critical functionality to the client) without any proper plan, you are going to test major functionalities.
Conditional release to the client
When we have more time, tester will do random testing on functionality & intention to break the applicatn functionality.

What is test plan & diff. sectn in it?
To plan our testing activities to testing from beginning to ending with clear dates with scope of testing.

What are risks and contigency (sol n of risk) planning?
Risks are unexpected problems getting in the project & contigency plan is the approach / plan to resolve the issues.

What are different type of applicatn we can test?
windows, client server, web, distributed, mobile applicatn

imp What is bug life cycle?
imp testing life cycle?

What is bug & defect remark?
Bug is raised by the testing team & when it is accepted by the dev team then calling it as defect.
Remark is bug rejected by the dev team.

What is deffered bug?
due to lack of time, Any defect is not fixing in current release & will be fixing in future release then dev team will give the status as deffered.

imp How the process in Agile model?

when Agile => for unstable requirements project, if there is any requirement change then

drawbacks in waterfall =>

1. No verificatn technique
2. when there is any change in Requirement again need to from begin
3. there is no bottom to up.
4. If requirements are unstable, time & budget will be very high.

what are the diff. sections we will fill while raising the bugs?

summary, description, steps to reproduce, project, modis, Sprint no., Severity, Priority, founded by, Assigned to, comments, States.

imp what is Severity & priority?

Severity is talking about seriousness of the issue & given by the testing team.

priority is talking about how much urgently to fix the issue & given by the developer.

what is diff. betn validatn & verificatn
↓
actual testing

Quality Assurance technique is verificatn & here we will verify all our documentatn is according to the requirement or not.

Quality control technique is validatn & here we will perform test execution & verify whether working according to the requirements or not.

low severity ⇒ cosmetic changes, GUR level issues
spell mistake, logo, -

High priority ⇒ logo, it is low severity
as per tester but as per client
it is high priority bug so developer will
fix it first.

low priority ⇒

high severity ⇒ if some functionality is not
working. but this functionality
is not that much imp for
client that can fix it later
also.

testing point of view imp ⇒ high severity
Client point of view imp ⇒ high priority

How many types of test cases you write?
GUI, functional +ve, functional -ve, field level
validation.

imp What is SDLC?

How many test cases will you write per day?
In my project, we are executing 30-40 test
cases having 10-15 steps in each test case.

What is usability testing?
test user friendly or not.

imp test case design technique

⇒ Sample test cases for any objects like pen,
water bottle, chair...

what is compatibility & end to end testing?
these are come under system level testing
here we will verify applicatn is working end to end on multiple platforms & with multiple h/w configuratn
→ how applicatn behaving in end to end / linux / windows or in browsers.

what are the metrics do you have ~~in~~ an idea?
~~A~~. metrics ⇒ measurement (done by lead, manager)
measure ⇒ ① Remark / Defect Ratio
② Test Coverage
③ Defect Severity Index

what is alpha & beta testing?
Alpha ⇒ In UAT testing, client is coming to the Company & doing the testing is Alpha.
Beta ⇒ he will do testing in his company environment

<u>Agile process</u>

PO responsibilities ⇒

Severity ⇒ impact of bug or defect on customer business workflow

if bug impact on workflow is less then severity will be less

critical ⇒ main functionality is not working. basic functionality itself is not working. customer workflow has been completely broken.

major ⇒ when you dont know how will be impact of bug on customer workflow.
for eg. if you will send mail but don't get notificat^n for that.

minor ⇒ 100% sure that bug will not be impact on business workflow
eg. look n feel, spell mistake, alignment

Priority ⇒ how soon bug should be fixed, how soon you want developer to fix that bug importance given to bug to fix it.

Priority ⇒ ① It is order in which developer should resolve a defect.

② order in which defect should be fixed. it is categorized into three types → high (urgent), medium, low

③ priority ~~is~~ indicates how soon the bug should be fixed.

④ priority is associated with scheduling.

⑤ priority is driven by business value.

⑥ priority value is subjective & can change over a period of time depending on the change in project situation.

⑦ importance of defect w.r. to client point of view ... means how soon it should be fix.

<u>Severity</u> ⇒ ① It is degree of impact ~~on~~ that a defect has on the operat^n of the product

② severity categorized into critical, major, moderate and minor. (importance of defect w.r. to functional point of view means criticalness of defect w.r. to application)

③ Severity is associated with functionality or standards.

④ Severity indicates the seriousness of the defect on the product functionality.

⑤ Severity levels of the defects are determined by QA engineer.

⑥ Severity is driven by functionality.

⑦ Severity value is objective & less likely to change.

High priority and low severity ⇒ indicates, defect have to be fixed on immediate bases but does not affect the applicat^n.

High severity and low priority ⇒ indicates defect have to be fixed but not ~~A~~ on immediate bases.

priority status is based on customer requirements where as severity status is based on the technical aspect of product

Verification	Validation
⇒ verification is to verifying the documents	⇒ validation is to validate the actual & expected o/p of the s/w
⇒ Activities involved here are Inspections, Reviews, walkthroughs.	⇒ Activities involved in this is Testing the s/w application
⇒ It is a low-level activity	⇒ It is high level Activity
⇒ It is a static method of checking documents & files.	⇒ Validatn is a dynamic process of testing the real Product.

Test Scenarios	Test Cases
⇒ It gives the idea of what we have to test.	⇒ Test cases are the set of positive & negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post condn & actual results.
⇒ It is like a high-level test case.	
what to be tested	How to be tested

S/w testing → Manual
→ Automatn

Severity ⇒ Bug/defect severity can be defined as the impact of the bug on customer's business.

critical, major, minor → imposes some loss of functionality.

critical ⇓ cannot move further is broken

major ⇓ lot of functionality

priority ⇒ Bug/defect priority can be defined as how soon the bug should be fixed.

high, low, medium

High severity High priority ⇒

Key features failed & no workaround

eg. login button is not working

low severity High priority ⇒

Basic feature failed but it has a huge impact on customer's business

eg. misspelled company logo.

low priority High severity ⇒

Key feature failed but there is no impact on customer's business.

eg. calculation fault in yearly report which end user wont use regularly.

low priority low severity ⇒

cosmetic issues

⇒ font family mismatch in a report.

Principle of SW Testing

① Early testing
verify requirement at early stage.

pect ② testing in order to find defects (testing to find defect)

③ It is highly impossible to give the error free SW to the customer. (Bug free SW not possible)

④ Should not do exhaustive testing (NO)
- doing same testing multiple time for eg.

Amt [↑]
100.62 then 9.01, 25.01 ← waste of time.

Same kind of input in multiple time

⑤ Testing is context based →
depending upon ~~time~~ type of application, you need to decide testing you will perform testing.

⑥ Should follow pesticide paradox (Says that)
if you are testing same set of test cases for the longer run those test cases won't be able to fix the defects
As time passes you have to change/update your test cases
after one or release, you have to think or update your test cases
to find more test cases (keep on)

⑦ Should follow ~~diff~~ defect clustering.
Some of the modules contains most of the defects across all
& those defect will not be uniformly distributed ~~in all~~
modules which will lead to operational failure
of the application.
focus on impacted area At early stage

SDLC → It is process of developing various quality s/w

What is S/W testing?

① S/W testing means identify defects in S/W
② test developed application, whether it is working according to client requirement or not.
③ It is a method to check whether the actual S/W product matches expected requirements & to ensure that S/W product is defect free.

Why testing is done?

Testing is done to assure the quality.
Quality is defined as Justificatn of all the requirements of a customer in an application.
Quality product ⇒ fullfill all the requirement of customer.

Objective of Testing ⇒

① Ensuring Quality, correctness, Robustness, Reliability

Robustness → critical function is working or not

Reliability → applicatn should be user friendly, accept by user

Adv of testing ⇒

① Quality product
② Client satisfactn
③ More business
④ Bug free product

SDLC → It is process of developing various quality sw

Benefits of Testing

① To identify the defects in development phase

② To ensure quality of the product (product quality)

— It is an essential requirement of any sw product. Testing ensures a quality product is delivered to customers.

③ Saves money as defect identified in earlier stages (cost effective)

[cost effective

④ Security ⇒

people are looking for trusted products.

it helps in removing risks & problems earlier.

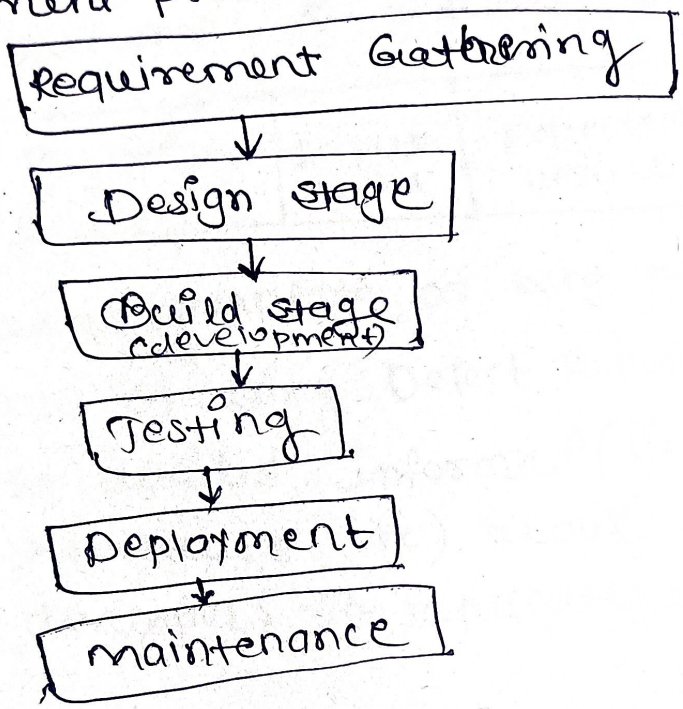
⑤ To build customer confidence & business (customer satisfaction) ⇒

The main aim of any product is to give satisfaction to their customers.

SDLC → It is process of developing various quality s/w

→ SDLC aims to produce a high quality s/w that meets customer expectation.

→ The life cycle defines a methodology for improving the quality of s/w & the overall development process.



A test case template is a document containing an organized list of test cases for diff. test scenarios that check whether or not the S/W has the intended functionality.

other than the executable steps, a test case also contains preconditions for testing, test data provided, expected outcome & actual result.

TC#	TS#	Test Design/ Steps	Input Data	Expected Result	Actual Result	pass	fail

defect report template or bug report template :-

The purpose of using Defect Report template is to convey the detailed informat^n (like environment details, steps to reproduce etc) about the bug to the developer. It allows developers to replicate the bug easily.

① Explain Agile methodology?

① Agile is an iterative development methodology. It promotes continuous iteration of development and testing throughout the s/w development lifecycle of the project.

② In the Agile model in s/w testing, both development & testing activities are concurrent, unlike the waterfall model.

③ Customer satisfaction by rapid delivery of useful s/w.

④ Welcome changes in requirements, even late in development.

⑤ Working s/w is delivered frequently (weeks rather than months)

⑥ Error can be fixed in the middle of the project

⑦ Development process is iterative, & the project is executed in short (2-4) weeks iterations. planning is very less.

⑧ Testers & developers work together.

① Scrum ⇒

Scrum is an agile development method which concentrates specifically on how to manage tasks within a team-based development environment.

Agile & scrum consist of three roles & their responsibilities are explained as follows:-

(i) Scrum Master → Master is responsible for setting up the team, sprint meeting & removes obstacles to progress.

The scrum master is a faciliator for the team & product owner.

② Product owner ⇒ ① The product owner creates product backlog, prioritizes the backlog & is responsible for the delivery of the functionality at each iteration

Person whom holds the vision for the product.

③ Scrum team → Team manages its own work & organizes the work to complete the sprint or cycle.

④ product Backlog → This is a repository where requirements are tracked with details on the no. of requirements (user stories) to be completed for each release.

It should be maintained & prioritized by product owner, & it should be distributed to the scrum team.

"A prioritized list of stories that are waiting to be worked on."

⑤ User story → A shorthand requirements document.

⑥ Sprint → A development process.
Each iteration of a scrum is known as Sprint.

⑦ sprint plan → It is taking care by project manager & scrum master based on particular plan.

Explain STLC

STLC is a sequence of different activities performed during the s/w testing process to ensure s/w quality goals are met.

→ STLC is a fundamental part of SDLC

STLC yields a step-by-step process to ensure quality s/w.

In the initial stages of STLC, while the s/w product or the applicatn is being developed, the testing team analyzes & defines the scope of testing, entry & exit criteria & also the test cases. It helps to reduce the test cycle time & also enhance the product quality.

As soon as the development phase is over, testing team is ready with test cases & start the execution. This helps in finding bugs in the early phase.

Phases of STLC →

① Requirement Analysis →

It is first step of s/w testing life cycle. In this phase quality assurance team understands the requirements like what is to be tested.

If anything is missing or not understandable then quality assurance team meets with the stakeholders to better understand the detail knowledge of requirement.

② Test Planning →

It is most efficient phase of s/w testing life cycle where all testing plans are defined. In this phase manager of the testing teams calculates estimated effort & cost for the testing work. This phase gets started once the requirement gathering phase is completed.

③ Test Case Development ⇒

It gets started once the test planning phase is completed. In this phase testing team note down the detailed test cases.

Testing team also prepare the required test data for the testing.

When the test cases are prepared then they are reviewed by quality assurance team.

④ Test Environment setup ⇒

It is the vital part of the STLC. Basically test environment decides the ~~conditions~~ SW or HW on which SW is tested.

This is independent activity & can be started along with test case development.

In this process the testing team is not involved. either the developer or the customer creates the testing environment.

⑤ Test Execution ⇒

After the test case development & test environment setup test execut^n phase gets started.

In this phase testing team start executing test cases based on prepared test cases in the earlier step.

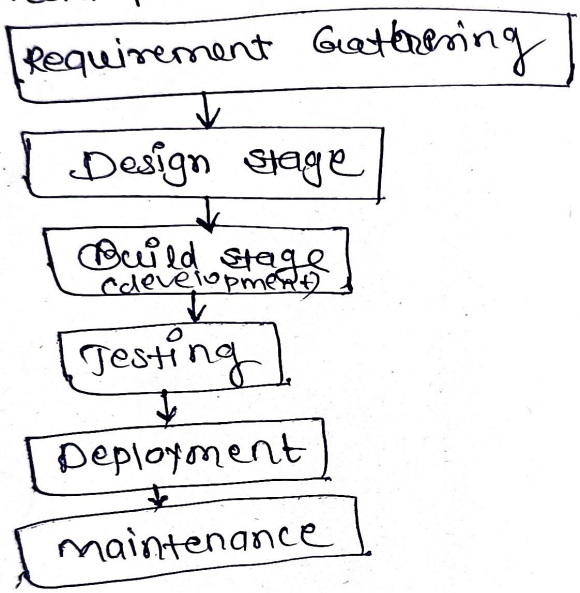
⑥ Test Closure ⇒

This is the last stage of STLC in which the process of testing is analyzed.

SDLC → It is process of developing various quality s/w

→ SDLC aims to produce a high quality s/w that meets customer expectation.

→ The life cycle defines a methodology for improving the quality of s/w & the overall development process.

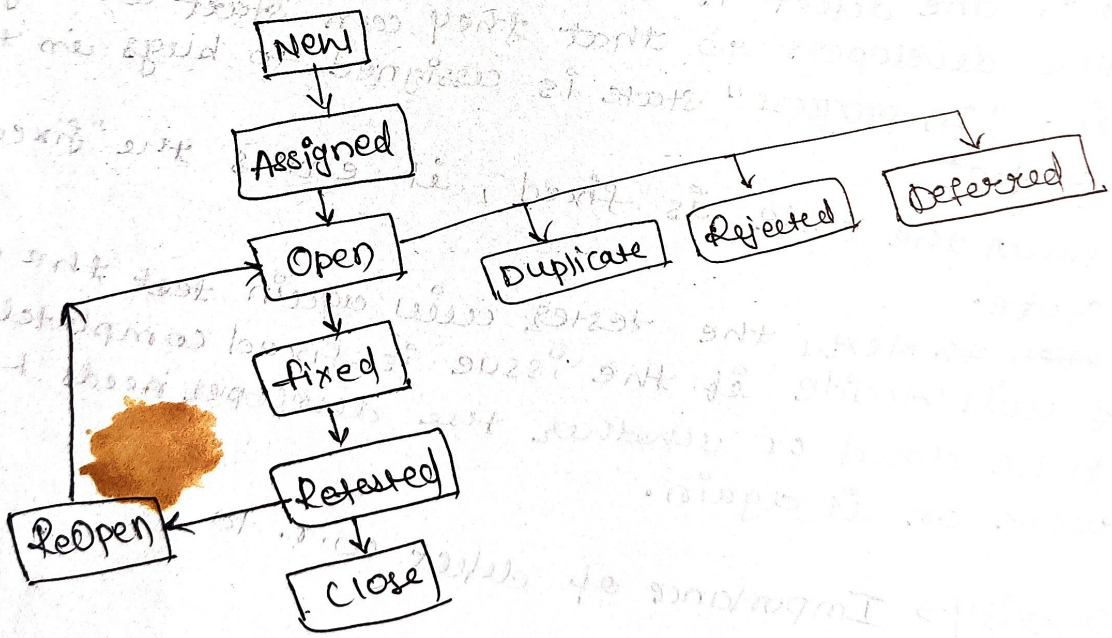


Bug life cycle / defect life cycle

① It is a process in which defect goes through diff. stages in its entire life.

② The Lifecycle starts as soon as a bug is reported by the tester & ends when a tester ensures that the issue is fixed & will not occur again.

③ The purpose of defect life cycle is to easily co-ordinate & communicate current status of defect & make the defect fixing process systematic & efficient.



How Does Defect Mangement lifecycle Works?

→ While testing the program, the tester finds a bug.

→ "New" status is assigned to the defects.

→ The defect is sent to the project manager so it can be analyzed properly.

→ The project manager makes the decision regarding whether it is valid or not.

→ If the issue is not valid, it is discarded by the project manager & "Rejected" status is assigned.

→ The manager also verifies that if such an issue was raised previously or not.

All these tasks are done through defect management tools for better understanding & transparency.

→ If the defect is new then it is assigned to the developers so that they can start working on it. "In-progress" state is assigned to bugs in this state.*

→ when the code is fixed, it enters the "fixed" state.

→ ~~when th~~ Next, the tester will again test the case & will decide if the issue is fixed completely to be closed or whether the developer needs to work on it again.

Severity → Importance of defect w.r. to

Agile

Scrum ⇒

→ process model

Agile ⇒ Iterative, & incremental model/Approach/process

agile is

① customer no need to wait for long time.

② We develop, test & release piece of s/w to the customer with few number of features.

③ we can accept/accomodate requirement changes

④ good communicat^n b/w^n customer, B.A, developers & testers.

Advantages ⇒

① requirement changes are allowed in any stage of development (or) we can accomodate requirement changes in the middle development.

② Releases will be very fast (weekly).

③ customer no need to wait for long time.

④ Good communicat^n b/w^n team

⑤ it is very easy model to adopt.

disadvantage ⇒

① less focus on design & documentat^n since we deliver s/w very faster.

Scrum ⇒

Agile is process model, defined principles (or process) (how the process ~~will~~ should be) Scrum is framework through which we will develope & test the SW. & then finally will release the SW.

⇒ To follow that process / principles, scrum is a kind of framework which will help us.

⇒ scrum ⇒ how to follow that process.

→ Scrum is a framework through which we build SW product by following Agile principles.

→ scrum team / Agile team ⇒ Some peoples are involved

Product owner, Scrum master, dev team, QA team.

① Product owner ⇒ writing requirement / feature

→ ① defined features of the product (always interact with customer) in the form of user story or epic.

② prioritize features according to market value.

③ Adjust features & priority every iteration, as needed

④ Accept or reject work results.

② Scrum master ⇒ He will responsible for that everyone follow process see there ~~is~~ any ~~bo~~ blocker is there or not.

→ The main role is facilitating & driving the agile process.

→ He will make sure that dev & testers team follow Agile process.

→ each & every process he will know

dev & QA ⇒ develope & test the SW

Scrum Terminology

① user story ⇒ A feature / module in a SW.
(small requirement)

② Epic ⇒ collection of user stories. (large requirem)
(huge requirement) split into small user story.

③ product Backlog ⇒ contains list of user stories, prepared by product owner at the beginning of process. (100 User story)
↓ document or excel
all stories in product backlog

④ Sprint / iteration ⇒ period/span of time to complete the user stories, decided by the product owner and team, usually 2-4 weeks of time.

⇒ ~~entire~~ complete design, development, testing entire process should be completed for particular feature.

In 2-4W deliver some working functionality

Sprint planning meeting ⇒ Meeting conducts with the team to define what can be delivered in the sprint & duration.
conduct at the beginning of stage of project where team will plan what are stories the the stories develope & test during sprint cycle

① How many stories are in backlog
② How many stories are going to develope and test during the sprint
③ duration of sprint.

Sprint Backlog ⇒ List of committed stories by DEV/QA for specific sprint.
⇒ only few no. of user stories for specific sprint.
⇒ subset of product backlog.

Scrum meeting ⇒ meeting conducted by scrum master everyday 15 mins called as standup meeting.

① what did you do yesterday?
② what will you do today?
③ Are there impediments in your way?

⇒ discuss any blocker
⇒ track entire process.

Sprint retrospective meeting ⇒ conduct once after completion of every sprint.
discuss about what went wrong, what improvement is needed, what went good/well. other improvement need to be done in upcoming sprints.
⇒ conducts meeting after completion of sprint.
The entire team including both the Scrummaster & the product owner should participate.

Story point ⇒ Rough estimation of user stories, will be given by dev & QA in the form of fibonacci series.

[on what basis they will select stories]
based on story point, we choose user story & start working on it.

0 - 1 1 2 3 5 8

1 = 1 hour / 1 day (-6-8 hours)

dev = 5 hrs, QA = 3 hrs ⇒ 8 hrs (1 day)

Burndown chart ⇒ Shows how much work remaining in the sprint.
↓ Status prepared by SM
maintained by the scrum master daily.

1 story ⇒ 3 days (18 hours).

Scrum process

> product owner always interact with customer to take i/p from stakeholder

Roles → product owner, scrum master, teams

Artefacts ⇒ product backlog, sprint backlog, burndown chart (document)

Ceremonies ⇒ sprint planning, daily scrum, sprint review.

Testing Course Agenda

Module 1 : Testing Fundamentals
Module 2 : Testing Roles and Responsibilities
Module 3 : SDLC & STLC Models
Module 4 : Test Management Team
Module 5 : Test Engineer Responsibilities
Module 6 : Defect Management
Module 7 : Status Reports
Module 8 : Automation

Overview of Software Testing

What is Software → Software is a set of instructions, data or programs used to operate computers and execute specific tasks

SDLC →
- Requirements gathering
- Analysis
- Design
- Coding/Development
- Testing
- Deployment/Production
- Maintains/ Support

Roles & Responsibilities
- Business analyst
- project manager
- Developers
- Testers
- Deployment/configuration Team

Software Testing → Testing an application working as Per client requirements

Manual Testing
- White box testing (Developers)
- Black box testing (Testers)
  - Functional Testing
  - Non-Functional Testing

Automation Testing
Tools
Selenium ,
QTP etc

STLC →
- Understanding requirements
- Test Plan
- Test Scenarios
- Test Cases
- Test Case Execution
- Defect Reporting
- Test Closer

Security Testing
Performance Testing

Examples : Smoke Testing ,Sanity Testing, Re-Testing ,Regression Testing etc

Sr. Software Test Engineer Responsibilities

- Same as test engineer responsibilities

+

- Participates in Review of Test Scenarios ,Test cases and defects
- Some Times involved in Test Plan preparation also.
- If required Leading the team when Team Lead is on Vacation

Test Lead Responsibilities

- Task Preparation and allocation to Team members
- Training Team members
- Team Management
- Test Scenarios & Test Cases Reviews
- Bug Reviews
- Preparation of Build summary report
- Conducting meetings with Team members
- MOM Preparation
- Test Plan preparation

Test Manager Responsibilities

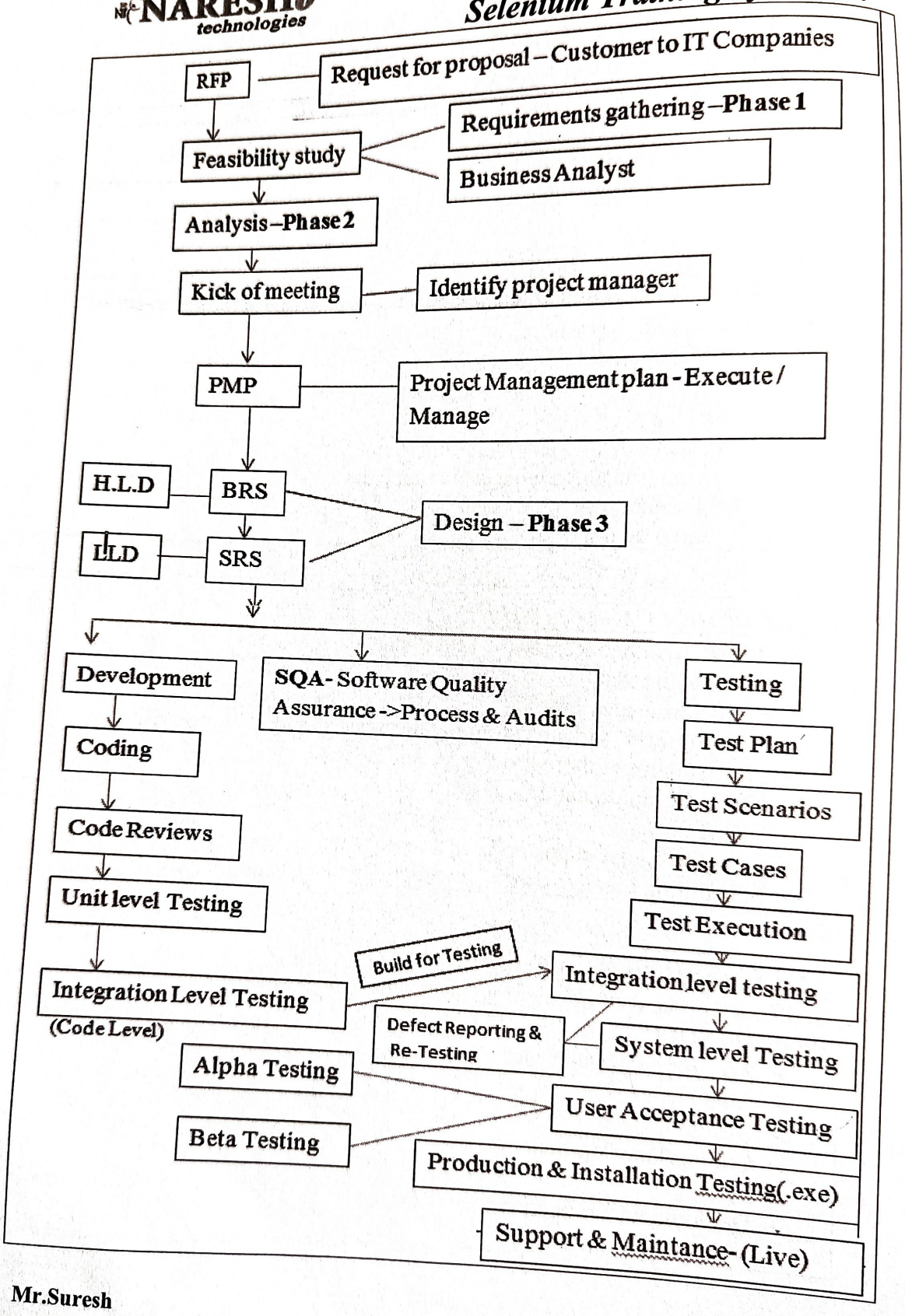
- Project plan and Review of Test Plan
- Effort estimates
- Project Management
- Training Plan – Identify training need based on Resource skills
- Preparing monthly reports
- Client Communications
- MOM
- Provide regular status updates to core team
- Scheduling meetings with Development and Testing Team

SDLC–Software Development Life Cycle

SDLC is a process of developing various quality Software.

Phases in SDLC :

- Requirements gathering
- Analysis
- Design
- Coding/Development
- Testing
- Deployment/Production
- Maintains/Support

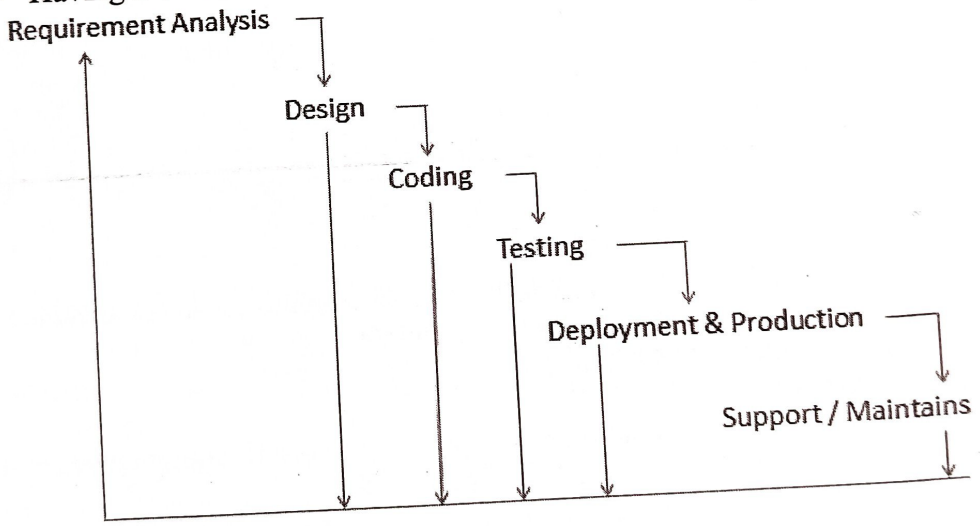

RFP
Request for proposal – Customer to IT Companies
Feasibility study
Requirements gathering –Phase 1
Business Analyst
Analysis –Phase 2
Kick of meeting
Identify project manager
PMP
Project Management plan - Execute / Manage
H.L.D
BRS
LLD
SRS
Design – Phase 3
Development
Coding
Code Reviews
Unit level Testing
Integration Level Testing
(Code Level)
SQA- Software Quality Assurance ->Process & Audits
Testing
Test Plan
Test Scenarios
Test Cases
Test Execution
Build for Testing
Integration level testing
Defect Reporting & Re-Testing
System level Testing
Alpha Testing
Beta Testing
User Acceptance Testing
Production & Installation Testing(.exe)
Support & Maintance- (Live)

Types of SDLC Models

- Waterfall model **or** Life Cycle Model **or** Linear Sequential Model
- V- Model **or** Verification & Validation Model
- Agile …etc

Waterfall model

- This model suggests a systematic and sequential approach to software development that begins at requirements analysis and progress through all life cycle phases sequentially
- Suitable for projects where requirements are clearly defined
- Small and medium term duration
- Having Domain knowledge



Advantages :

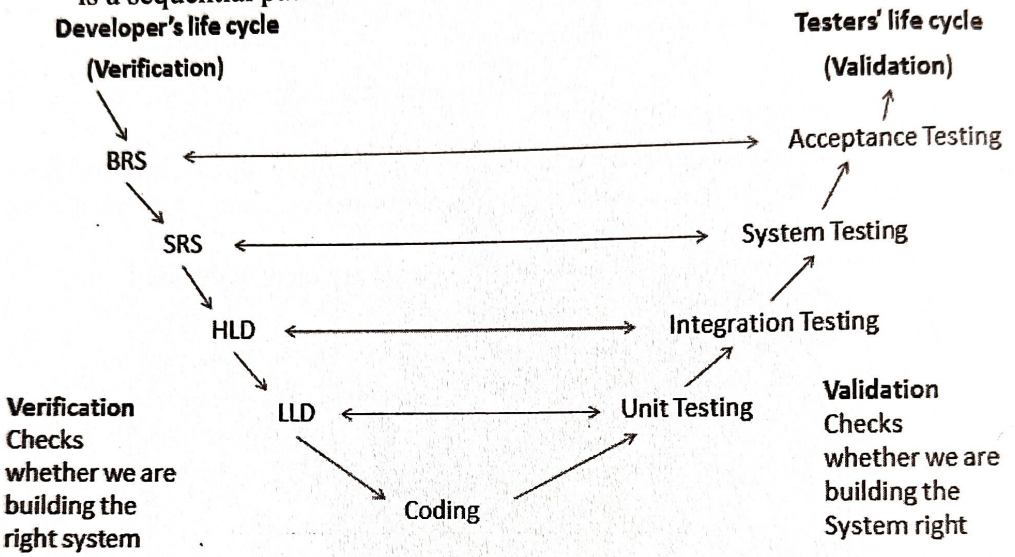
- Project under Control
- Pre-defined outputs at every phase
- Tracking changes is easy

Disadvantages:

- Not suitable for requirements changes
- Does not support going back to previous phase
- If any defect found need to go back to the originating phase

V - model

- V model means verification and validation model, the V shaped life cycle is a sequential path of execution process.



Advantages :

- Simple and easy to use
- Testing activities like planning and test design will be done before coding
- Testing planned parallel to development
- Bug detection in early phases

Disadvantages:

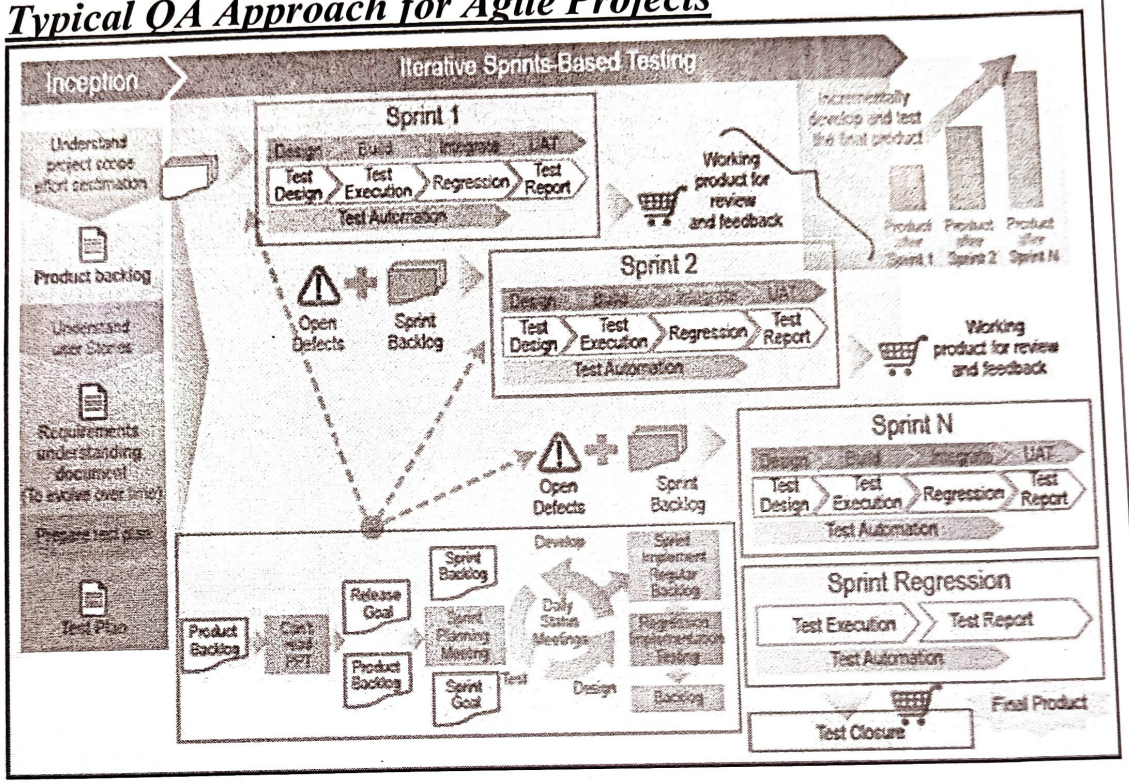
- If any changes happen in midway, then the test document along with requirements documents has to be updated.
- Not Suitable for large and complex projects
- The client sees the only final project ,not intermediate modules

Agile - Process

Overview of Agile Process

- Customer satisfaction by rapid delivery of useful software.
- Welcome changes in requirements ,even late in development
- Working software is delivered frequently (weeks rather than months)
- Close ,daily cooperation between business professionals and developers
- Continuous attention to technical excellence and good design

Typical QA Approach for Agile Projects

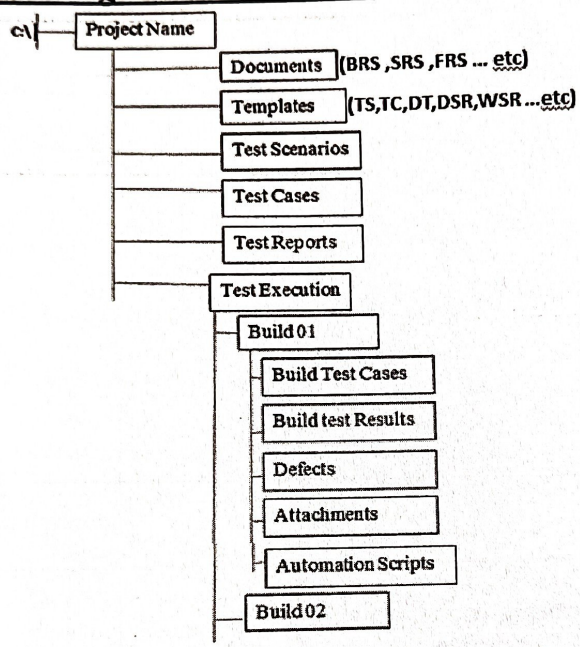


Agile - Terms

- **User Story** – A shorthand requirements document.
- **Product Backlog** -- A prioritized list of stories that are waiting to be worked on.
- **Product Owner** -- person whom holds the vision for the product.
- **Scrum Master Role** The Scrum Master is a facilitator for the team and product owner.
- **Sprint** -- A development process
- **Stand-up Meeting**– a short (15 minutes or less) daily meeting during which team members report on what they have accomplished since the last meeting, what they plan to accomplish today and report any impediments or blockers to making progress.
- **Scrum Meeting** – To discuss on Sprint planning, development and Review.

STLC → It is a sequence of specific activities conducted during the testing process to ensure s/w quality goals are met.

Working Folder Structure



STLC – Software Testing Life Cycle

- Requirements understanding (BRS, SRS, FRS, Mock-ups)
- Test Plan
- Test Scenarios
- Test Cases
- Test Execution
- Bug Reporting & Re-Testing
- Test Closer

Requirement understanding :- test teams studies requirements from a testing point of view to identify testable requirements.

Activity → ① Identify types of tests to be performed
② gather details about testing priorities & focus
③ Analyse business functionality

Test Plan

Test Plan – A document describing the scope, approach, resources & schedule of testing activities. It identifies test items ,the features to be tested, the testing tasks who will be doing and any risks …etc

Senior quality analyst, manager determines the test plan strategy with efforts and cost estimates

Entry Criteria to prepare Test Plan :

- Approved PMP
- Approved SRS
- Test Plan guidelines
- Test Plan template

Exit Criteria of preparing Test Plan :

- Test Plan should be reviewed and approved.

Table Contents in Test Plan

1. Introduction
  1.1 Test plan objectives
2.Scope of this document
3.Test Strategy
  3.1 Smoke Testing
  3.2 Sanity Testing
  3.3 Functional Testing
  3.4 Database Testing
  3.5 Cross browser Testing
  3.6 Automation Testing
4.Environment Requirements
5.Test Schedule
6.Control Procedures
  6.1 Reviews
  6.2 Bug Reviews
7.Functions To be tested
8.Functions not to be tested
9. Resources & Responsibilities
10. Deliverables and mile stones
11. Defect Management
12. Dependencies
  12.1 personal dependencies
  12.2 Software dependencies
  12.3 Hardware dependencies
  12.4 Test Data & Database
13. Risks
14. Tools
15. Documentation

Activity :-

1. test tool selection
2. test effort estimation
3. preparation of test plan for various types of testing
4. finalize on the best suit approach
5. Resource planning & determining roles & responsibilities.

16. Approvals
17. Entry / Exit for each Testing activity
18. Test Suspension
19. Test Resumption
20. Test completion

Test Scenarios → A test scenario is any functionality of the application under test, that can be tested.

test condn or test possibility

Test Scenarios: Identify all possible areas to be tested **or** What is to be tested.

Entry Criteria to Identify Test Scenarios :

- Approved Test plan
- Approved SRS
- Test Scenarios Guidelines
- Test Scenario template

Exit Criteria for Test Scenarios :

- Test Scenarios should be reviewed & approved(mapping test scenarios with requirements)

Test Scenarios Template

Company Logo			Project ID		Reviewed By	
			Project Name		Date Reviewed	
			Identified by		Approved By	
			Date Identified		Date Approved	
TS#	Req#	Main Functionality	TestScenario Description	Test Case Name/TC ID	Document ID/Reference	Comments

Test Case - ~~step~~ set of step by step instruction to verify that the system meets functionalities & works properly

Test Case: A test case is a document which contain set of input values, pre-conditions, expected result … etc **or** How to be tested.

Entry Criteria to Identify Test Cases :

- Approved Test plan
- Approved SRS
- Approved FRS
- Approved Test Scenarios
- Test Case Guidelines
- Test Case Template

Activity ⇒ ① Create test cases, test design, automation scripts
② Review & baseline test cases & scripts
③ create test data

Exit Criteria for Test Cases :

- Test Cases should be reviewed & approved(mapping test scenarios with requirements)

Test Case Template

		Project History				Test Cases				
Company Logo		Project ID				Total No. of TCs				
		Project Name				Passed				
						Failed				
		Test Case History				Not Executable				
		Created By		Date Created		Defects Reported				
		Reviewed By		Date Reviewed						
		Approved By		Date Approved						
		Test Information								
		Test Executed By		Date Executed						
		Version		Build						
TC #	TS#	Test Design/ Steps	Input Data	Expected Result	Actual Result	Pass	Fail	Not Executable	Comments	Defect Id

Test Case Design Techniques

Test Case: Design techniques can broadly split in to 2 categories.

1. Black box techniques
2. White box techniques

Black box techniques

- Equivalence Class Partitioning(ECP)
- Boundary Value Analysis(BVA)
- State Transition
- Decision Table / Cause Effect Table

White box techniques

- Statement Testing
- Branch/Decision testing
- Data flow Testing
- Branch condition testing

Test Environment setup ⇒ ① It decides s/w & h/w condn under which a work product is tested.

Equivalence Class Partitioning (ECP) :

Divide a set of test conditions into groups which is having similar behavior to reduce number of test cases.

Age [ ] **(accepts 1 to 60)**

Invalid	Valid	Invalid
0	1 to 60	>=61

Boundary Value Analysis(BVA):

BVA is based on testing the boundaries of condition

Formula : Minimum ,maximum ,Min-1 ,Max+1

Valid Boundaries : Minimum , Maximum

Invalid Boundaries : Min-1 ,Max+1

Name [ ] **(accepts 5 to 10 characters)**

Invalid (Min-1)	Valid (Min,Max)	Invalid (~~Min~~ max+1)
4	5 , 10	11

State Transition Table:

Application provides different output for same input based on previous stage

Username	Password	Result
Correct	Wrong	Invalid Password
Correct	Wrong	Invalid Password
Correct	Wrong	Invalid Password
Correct	Wrong	Account Locked

Decision Table :

Testing with different combination of inputs which produce different results.

Username	Password	Result
Correct	Correct	Login Successful
Correct	Wrong	Invalid Password
Wrong	Correct	Invalid Username
Wrong	Wrong	Invalid Username

Test Execution → Real time validation of product & finding bugs. The process consist of test scripts execution, testscripts maintenance & bug reporting.

Test cycle closure → This phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices & test results.

Defect Management

Defect Status : New ,Open ,Fixed ,closed ,Re-Open ,Not a bug, Duplicate ,Need More information, Can't reproduce .
Severity: Importance of defect with respective to functional point of view…means criticalness of defect with respective to application.
Severity classification could be : CRITICAL , HIGH ,MEDIUM ,LOW
Priority: Importance of defect with respective to client point of view … means how soon it should be fix.
Priority Classification could be : P1-Urgent ,P2-High ,P3-Medium ,P4-Low

Difference between Defect /Bug/Error/Failure

Defect : Problem which is identified on Developer machine at development phase
Bug : Problem which is identified on Testers machine at testing phase
Error: Problem which is related to coding.
Failure : Problem which is identified By end users at production phase

Bug / Defect Template

Defect ID		Assigned To				
Status		Browser				
Severity		Found in Version				
Priority		Found in Build				
Module		Fixed in version				
Reported By		Fixed in build				
Title						

Description

Steps To Re-Produce :

1.
2.
3.

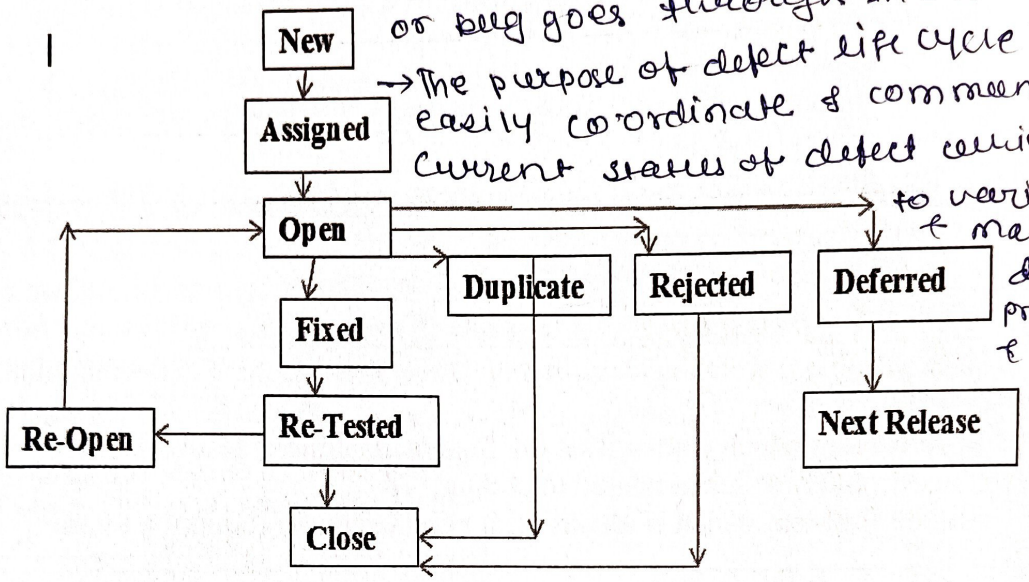
Expected Result :

Actual Result :

Bug or Defect Life Cycle

⇒ Defect life cycle in sw testing is the specific states that defect or bug goes through in its entire life.

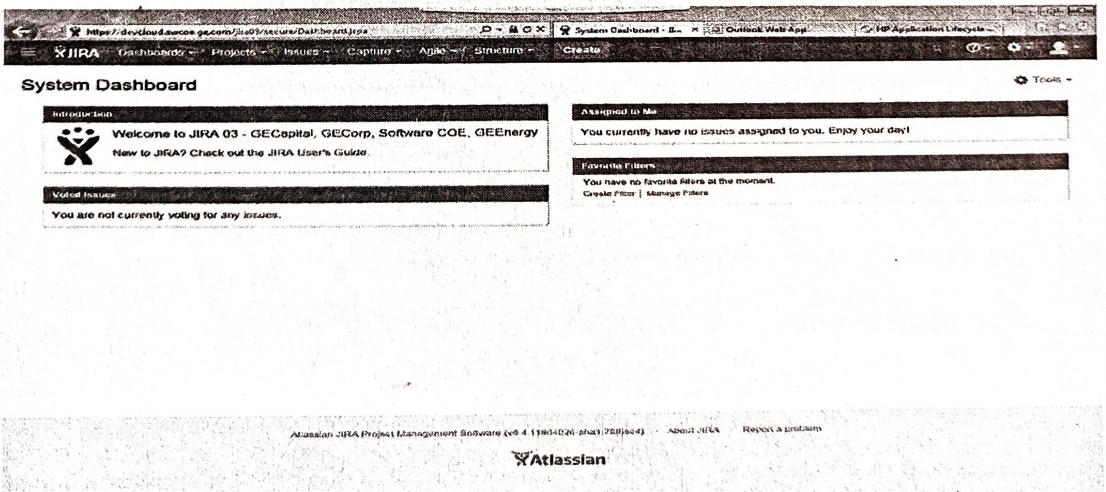
→ The purpose of defect life cycle is to easily coordinate & communicate current status of defect which changes to various assigns & make the defect fixing process systematic & efficient.



JIRA Tool:

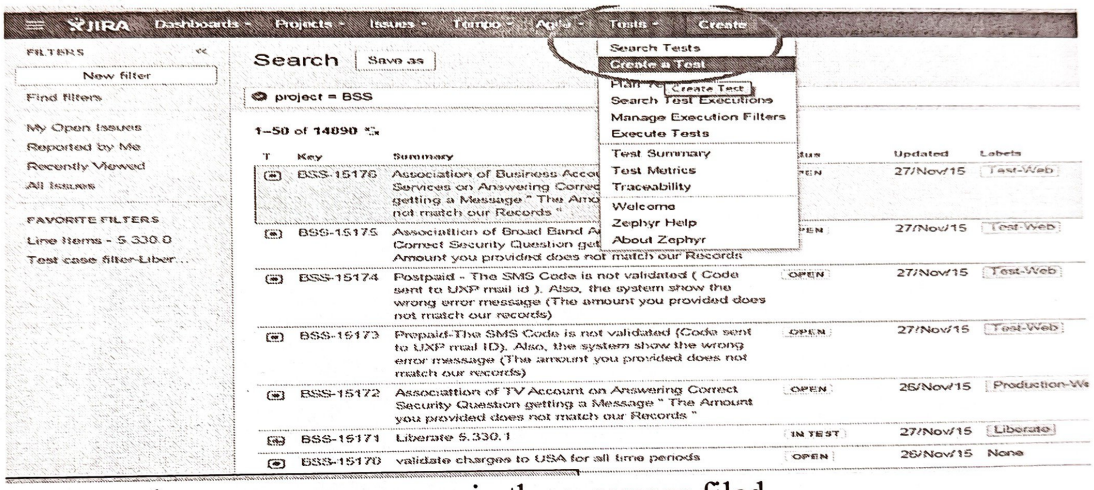
Jira Login Process:

- Open JIRA URL in the browser:
- Enter valid credentials (User Name and Password)
- JIRA Home Page shown as below:

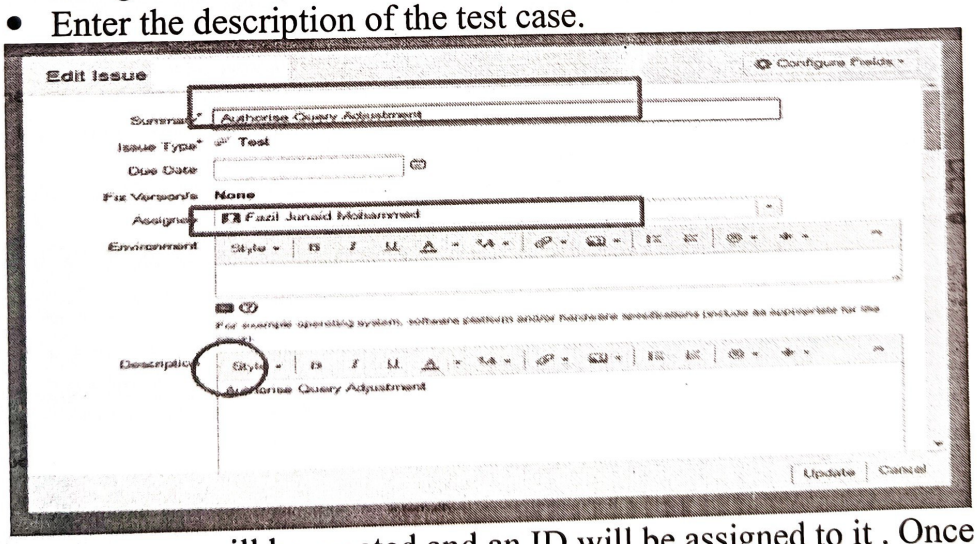


Steps to Create Test Cases in JIRA :

- In JIRA, Test case, Bug, Epic and Story will be called as Issues.
- To create a JIRA issue, you need the Create Issue project permission for the issue's relevant project. If you do not have this permission, please contact your JIRA administrator.
- To create a new JIRA issue (Test case):
- Click Create at the top of the screen to open the Create Issue dialog box.
- Select the relevant Project and Issue Type on the Create Issue dialog box.
- Under the tests tab, click on create a test.
- Please see the below screenshots for reference:



- Enter the test case summary in the summary filed.
- Assign the test case to anyone by typing their name.
- Enter the description of the test case.



- Test case will be created and an ID will be assigned to it . Once the test case is created you can add test steps and expected results as shown in next page.
- Click under the test steps and enter the steps. Click under the Expected results and enter the results and click on update.

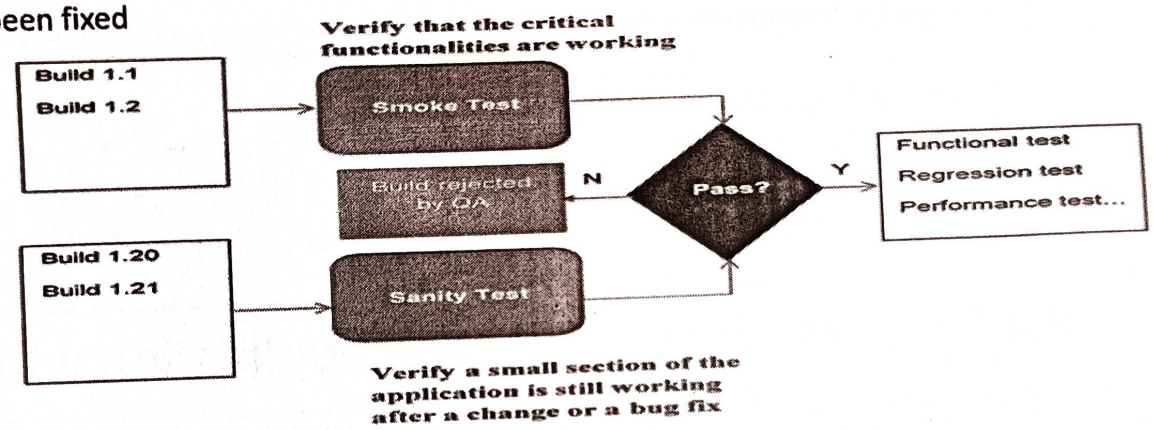
Types of Testing

- **Smoke Testing or BVT-Build Verification Testing or TAT – Testers Acceptance Testing**

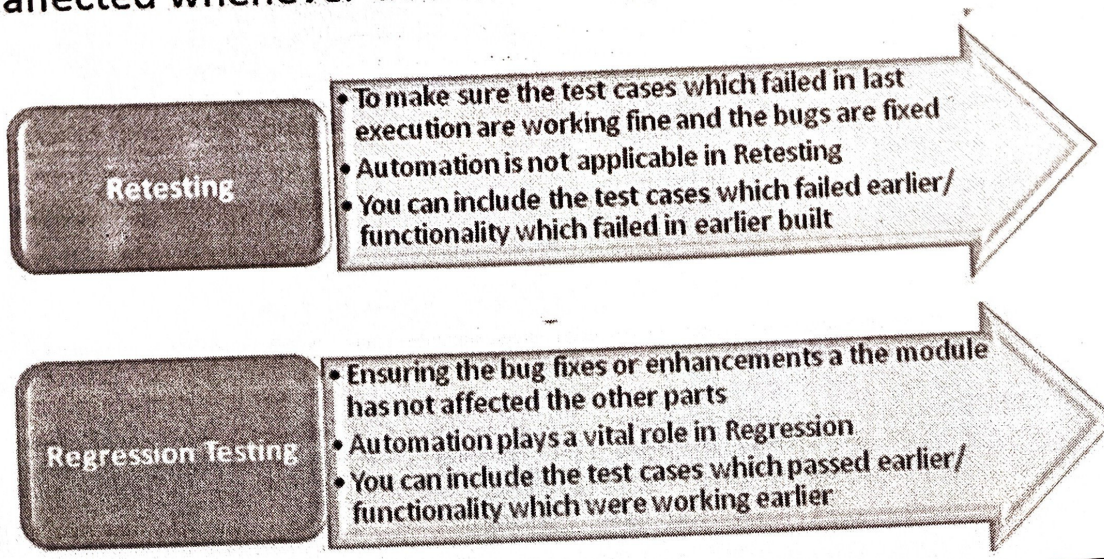
It is first level of testing on any newly released build to check main functionality of the application.

- **Sanity Testing :**

Sanity Testing is done during the release phase to check for the main functionalities of the application.Sanity Testing is done to check the new functionality/bugs have been fixed



- **Re-Testing :** Testing defects were fixed or not in the current build
- **Regression Testing:** To check existing functionality is un affected whenever the new change is added



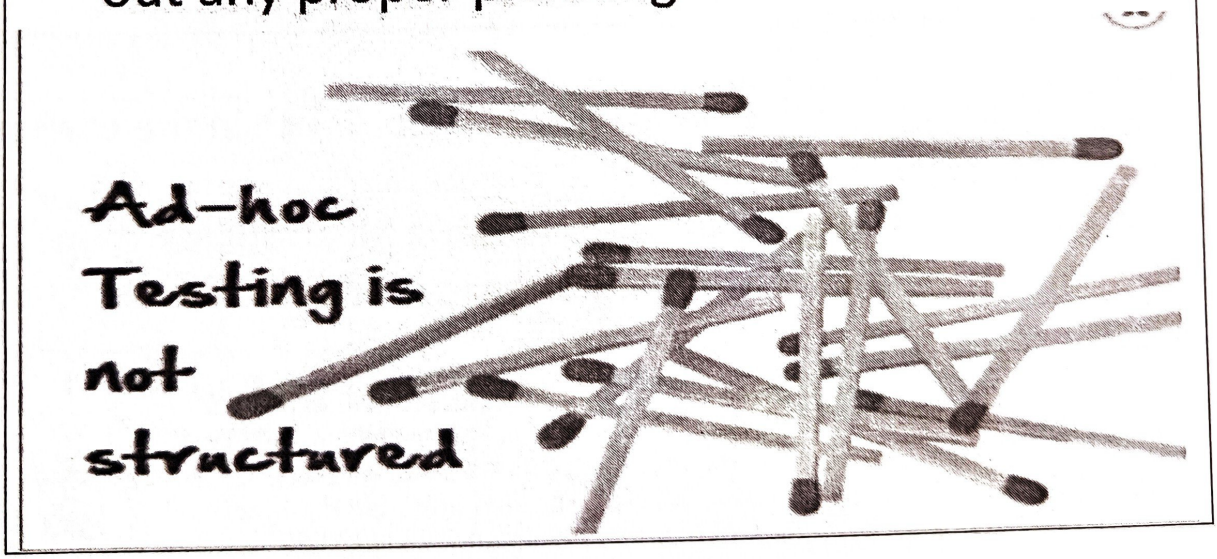
- **Static Testing:** Testing an application with out performing any action.

Eg: GUI Testing , colors ,spelling ,alignment ... etc.

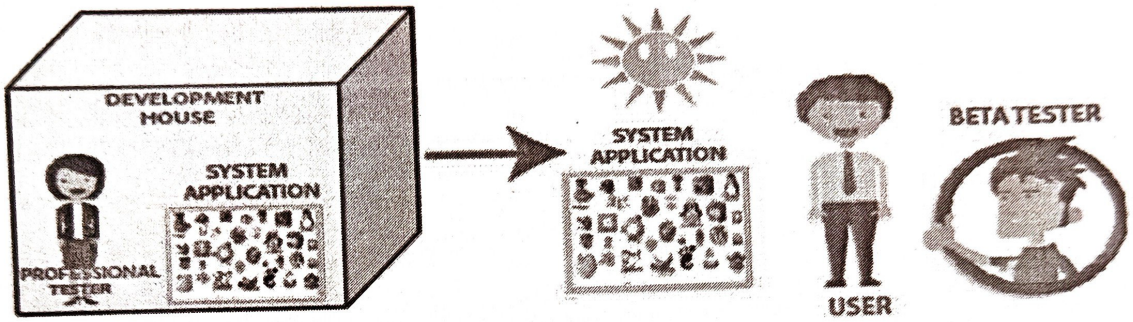
- **Dynamic Testing:** Testing an application by performing required actions.

Eg: Functionality Testing ,Textbox, button ... etc

- **Ad-hoc Testing:** Testing the application with out any proper planning



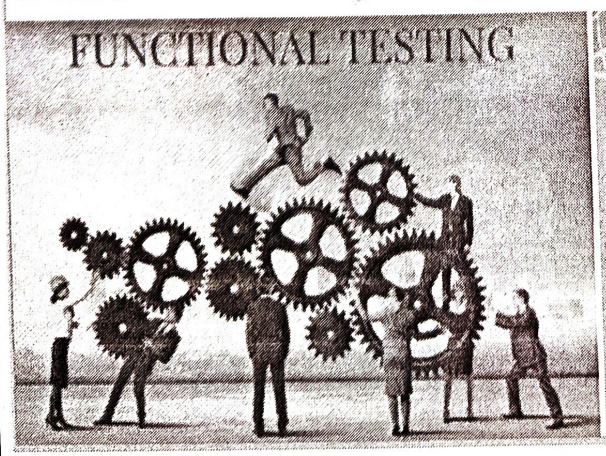
- **Alpha Testing :**Final Testing on the application doing with in the development company
- **Beta Testing:** Testing doing in customer environment ,This testing will be done by the customer or third party test engineers.

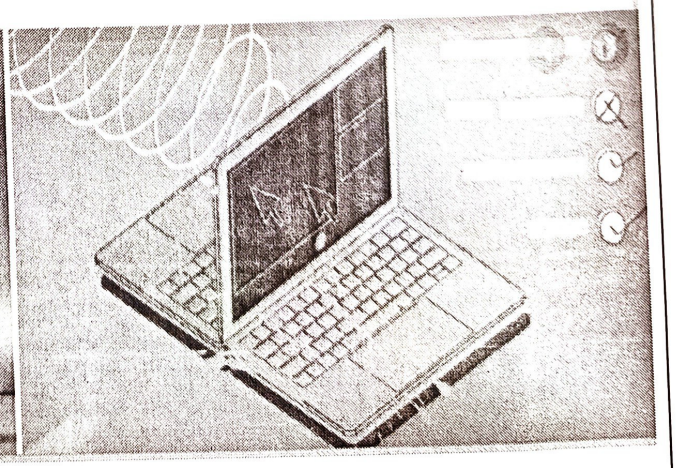


BETA TESTING OF THE PRODUCT IN REAL WORLD ENVIRONMENT

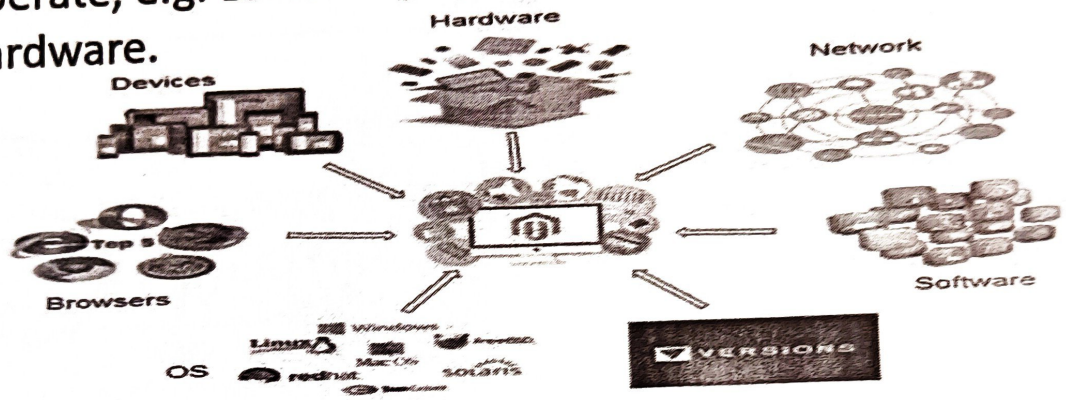
Functionality Testing - Test for – all the links in web pages, database connection, forms used for submitting or getting information from the user in the web pages, Cookie testing etc.

Usability testing - Usability testing is nothing but the User-friendliness check. In Usability testing, the application flow is tested so that a new user can understand the application easily. Basically, system navigation is checked in Usability testing.





Compatibility testing - Compatibility testing is used to determine if your software is compatible with other elements of a system with which it should operate, e.g. Browsers, Operating Systems, or hardware.



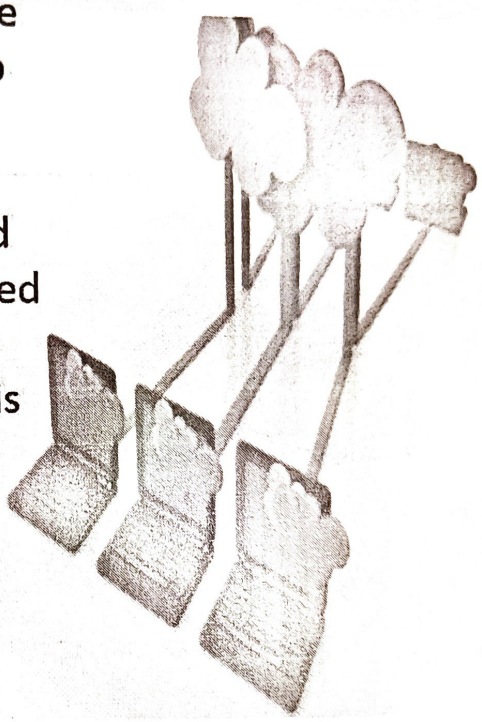
Database Testing – Database connection and user entered Data in application is saving into respective database tables

Interface testing-Three areas to be tested here are - Application, Web and Database Server

01. Application: Test requests are sent correctly to the Database and output at the client side is displayed correctly.

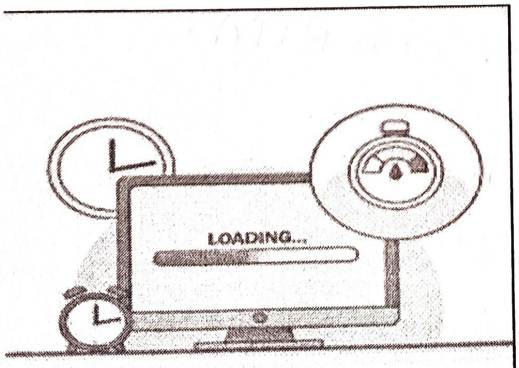
02. Web Server: Test Web server is handling all application requests without any service denial.

03. Database Server: Make sure queries sent to the database give expected results.

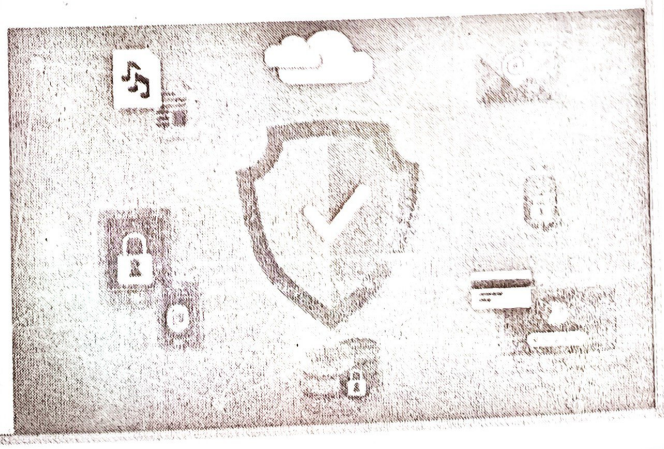


Performance testing – Testing page ,data, images load time

Security testing -Security Testing involves the test to identify any flaws and gaps from a security point of view.



Page Loading Speed



When to stop Testing?

This can be difficult to determine. Most modern software applications are so complex, and run in such an interdependent environment, that complete testing can never be done. Common factors in deciding when to stop are:

- Deadlines (release deadlines, testing deadlines, etc.)
- Test cases completed with certain percentage passed
- Test budget depleted
- Coverage of code/functionality/requirements reaches a specified point
- Bug rate falls below a certain level
- Beta or alpha testing period ends

Testexecutn → ① Real time validatn of product & finding bugs.

② The process consists of test script executn, test script maintenance & bug reporting.

③ If bugs are reported then it is reverted back to development team for correctn & retesting will be performed.

Activity ⇒ ① Execute tests as per plan

② document test results & log defects for failed cases

③ update test plans | test cases, if necessary.

④ map defects to test cases in RTM.

⑤ Retest the defect fixes.

⑥ Regression testing of applicatn

⑦ track the defects to closure.