

Manual Testing

→ Testing fundamental
SIW testing
SDLC
STLC

→ Manual Testing

Smoke testing
Sanity testing
Retesting
Regression testing
functional testing
Alpha, beta testing

→ Test case development

Test scenario
Test case
RTM

→ Test techniques

BVA
EPT
PTT
STT

→ Defects

Bug life cycle
Bug report
Severity & priority

→ Test management & control

→ practice
→ project

SIW ⇒ SIW is a set of instructions, data or programs used to operate computers and execute specific tasks.

Why testing job?

- ① SIW testers are made for challenging work environments.
- ② you can enjoy every day of work.
- ③ flexible & fun work environment
- ④ It is creative
- ⑤ It is a secure career path
- ⑥ There is attractive remuneration & room for growth.

Test Scenario \Rightarrow

(1)

- A test scenario is any functionality of the application under test, that can be tested.
- It is also called Test cond'n or test possibility

Testing Fundamentals

- ① What is SW testing?
- ② Why SW testing is imp?
- ③ Benefits of SW testing?
- ④ Why do you want to choose testing?
- ⑤ Types of testing & explain it
- ⑥ SDLC
- ⑦ Waterfall, V model, Agile
- ⑧ STLC
- ⑨ MT & AT basics
- ⑩ Types of testing with real time eg.

SQA quality is defined as a field of study & practice that describes the desirable attributes of SW product.

- Objective of testing ⇒
- ① Ensuring quality
 - ② Ensuring correctness
 - ③ Ensuring Robustness
 - ④ Ensuring Reliability

application should
be user friendly

accept by user

critical function
is working or not

Why testing is done?

Testing is done to assure the quality.
Quality is defined as justification of all the requirements of a customer in an application.

Quality product ⇒ fulfill all the requirement of customer.

Need for SW testing ⇒

- ① Delivering quality SW product.
- ② Satisfying user requirements.
- ③ making the SW more reliable
- ④ reducing maintenance cost.
- ⑤ avoiding customer dissatisfaction.

Adv of testing ⇒

- ① Quality product
- ② Client Satisfaction
- ③ more business
- ④ Bug free product

① What is SW testing?

- SW testing means identify defects in SW.
- It is a method to check whether the actual SW product matches expected requirements and to check/ensure that SW product is defect free. test developed application, whether it is working according to client requirement or not.
- It involves execution of SW or system components using manual or automated tools to evaluate one or more properties of interest.
- The purpose of SW testing is to identify the errors.
- SW testing is a process of verifying a computer system/ program to decide whether it meets the specified requirements & produces the desired results; As a result, you identify bugs in SW project.

Why SW testing is imp?

① It is imp because SW bugs could be expensive or even dangerous. So, it can be identified early & can be solved before delivery of the product.

② Properly tested SW product ensures reliability, security, and high accuracy. which further results in performance which further results in time saving, cost effectiveness & customer satisfaction.

if ask eg → In May of 1996, a SW bug caused the bank accounts of 823 customers of a major U.S. bank to be credited with 920 million US dollars. in development phases.

- ① To identify the defects
- ② To ensure quality of the product.
- ③ Saves money as defect identified in earlier stages.
- ④ To build customer confidence & business.

Benefits of SW testing ?

→ cost effective :-

testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of SW testing, it costs less to fix.

→ security :-

people are looking for trusted products. it helps in removing risks of problems earlier.

→ product quality :-

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

→ Customer satisfaction :-

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

types of testing

types of SW testing

Manual

white Box

Black box

there are
many
blocks of
testing

Automation

Grey Box

functional

unit testing

integration

system

user acceptance

Non-functional

compatibility
testing

performance

usability

carried out to
improve quality
& performance

FT \Rightarrow functions are tested
by providing some ZTP
& OIP compared to
expected result

Regression testing \Rightarrow Testing of already tested
system after modificat'

② Ensures that new code doesn't affect
the existing code.

SDLC
various stages
of SDLC

SDLC

→ It is a process of developing various quality sizes.

- ① first of all, RFP (Request for proposal) document provide by customer to IT companies.
 - ② Based on requirement proposal document we need to do feasibility study, here we need gathering feasibility study, we need to do requirement analysis.
 - ③ Based on requirement analysis this is 1st phase of SDLC. After requirement analysis will be done by subject matter expert (SME) / BA. After that there will be kick off meeting which is nothing but project confirmation done by BA, project manager, lead etc.
- (2) After that project manager will play very imp role here, he will come up with document PMP (project management plan) based on PMP document we need to get BRS (business requirement specification).

Based on BRS, we need to get SRS
(SIN Requirement Specification)

where as BRS is high level document
& SRS is low level document
this all comes under in design phase

after design phases, development and
testing need to be started.

⑤ Based on SRS, development and
testing need to be started.
in Repository (common storage) - SRS

document will uploaded.
then developer will start
development, they will follow
the process like

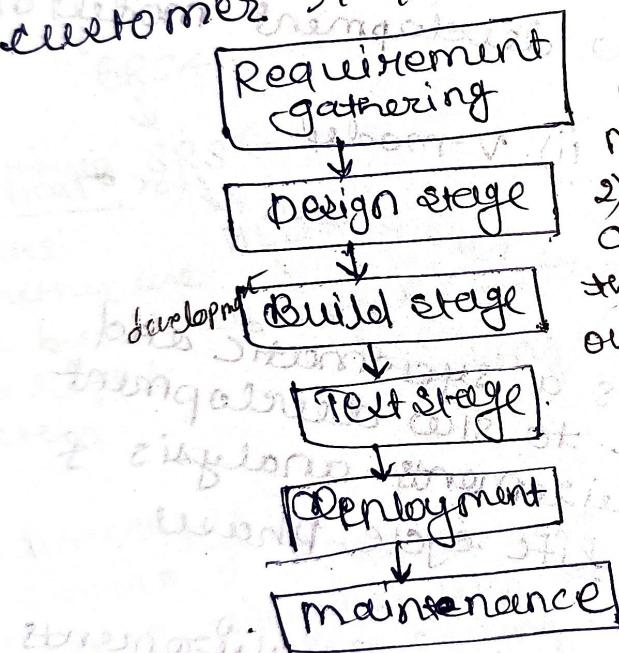
- ① Coding
- ② Code Review
- ③ unit level testing
- ④ integrated level testing

⑦ parallelly tester will do
requirement understanding

- ① test plan
- ② test Scenarios
- ③ test cases
- ④ test execution
- ⑤ Bug Reporting

Itation documents
to get SRS

- ⑧ Deployment → deploy the application in the respective environment
- ⑨ maintenance stage → once your system is ready to use, you may require to change the code later on as per customer request.



- 1) SDLC aims to produce a high quality SW that meets customer expectations
- 2) The life cycle defines a methodology for improving the quality of SW & the overall development process

SRS consist of all the product requirements to be designed & developed during the project lifecycle where the core is taken the process where the product is known as for the develop the product is known as maintenance.

SDLC Models

- To implement the SDLC process we need to understand SDLC models. There are various SDLC development models or methodologies:-
- i) waterfall model
 - ii) V-model
 - iii) Agile model

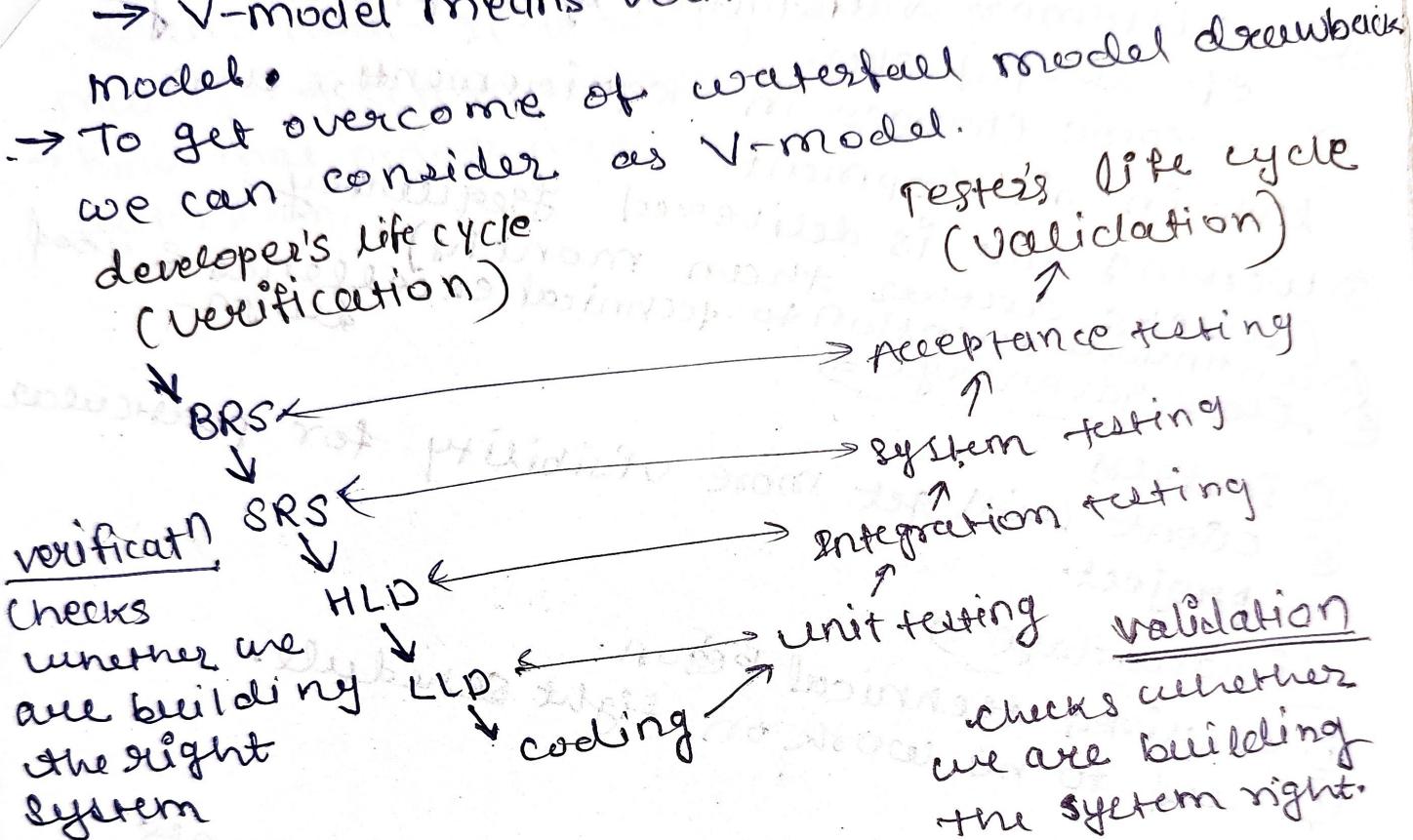
① Waterfall model :-

- This model suggests a systematic and sequential approach to SDLC development that begins at requirements analysis & progresses through all life cycle phases sequentially.
- Suitable for projects where requirements are clearly defined.
- Adv ⇒ ① project need to be small
② Requirements need to be very clear.
③ Pre-defined O/P at every phase.
- disadv ⇒ ① Developers & testers can't work together
② If any defect found need to go back to the Originating phase. One phase has to be totally completed then only we can start with next phase.
③ Doesn't support going back to previous phase.
④ Not suitable for requirement changes.

1 V-model
2 V-model
3 V-model
4 V-model

② V-model

→ V-model means verification and validation



Advantages =

- ① simple & easy to use.
- ② Testing activities like planning & test design will be done before coding.
- ③ Testing planned parallel to development.
- ④ Bug detection in early phase.

Disadvantages ⇒

- ① If any changes happen in midway, then the test document along with requirements documents has to be updated.
- ② Not suitable for large and complex projects.
- ③ The client sees the only final project not intermediate modules.

③ Sprint
management

Agile Model

- ① customer satisfaction by rapid delivery of useful SW.
 - ② welcome changes in requirements, even late in development.
 - ③ working SW is delivered frequently (weeks rather than months)
(continuous attention to technical excellence & good design)
 - ④ Close Advantage ⇒
 - ① fastness
 - ② client will get more visibility for particular project.
- disadvantage ⇒
- ① required technical person
 - ② need to work on tight schedule.

Agile Terms

- ① User story ⇒ ① A shorthand requirements document.
② low level requirement.
- ② product backlog ⇒ ① A prioritized list of stories that are waiting to be worked on.
② high level document.
- ③ product owner ⇒ person whom holds the vision for the project.
- ④ Scrum master Role ⇒ He is a mediator b/w team and product owner.

⑤ Sprint → A development process.
Sprint plan → It is taking care by project manager & scrum master based on particular plan.
→ how that process need to be done will come up with plan.



STLC [Software test life cycle]

→ STLC is a sequence of specific / diff. activities performed during the testing process to ensure SW quality goals are met.

→ STLC stands for SW testing life cycle.

STLC phases)

There are 6 major phases in every STLC

- ① Requirement Analysis
- ② Test planning
- ③ Test case development
- ④ ~~Test environment setup~~
- ⑤ Test execution
- ⑥ Test cycle closure.
- ⑦

- ① Requirement understanding
- ② Test plan
- ③ Test scenarios
- ④ Test cases
- ⑤ Test execution
- ⑥ Bug Reporting
- ⑦ Test closure.

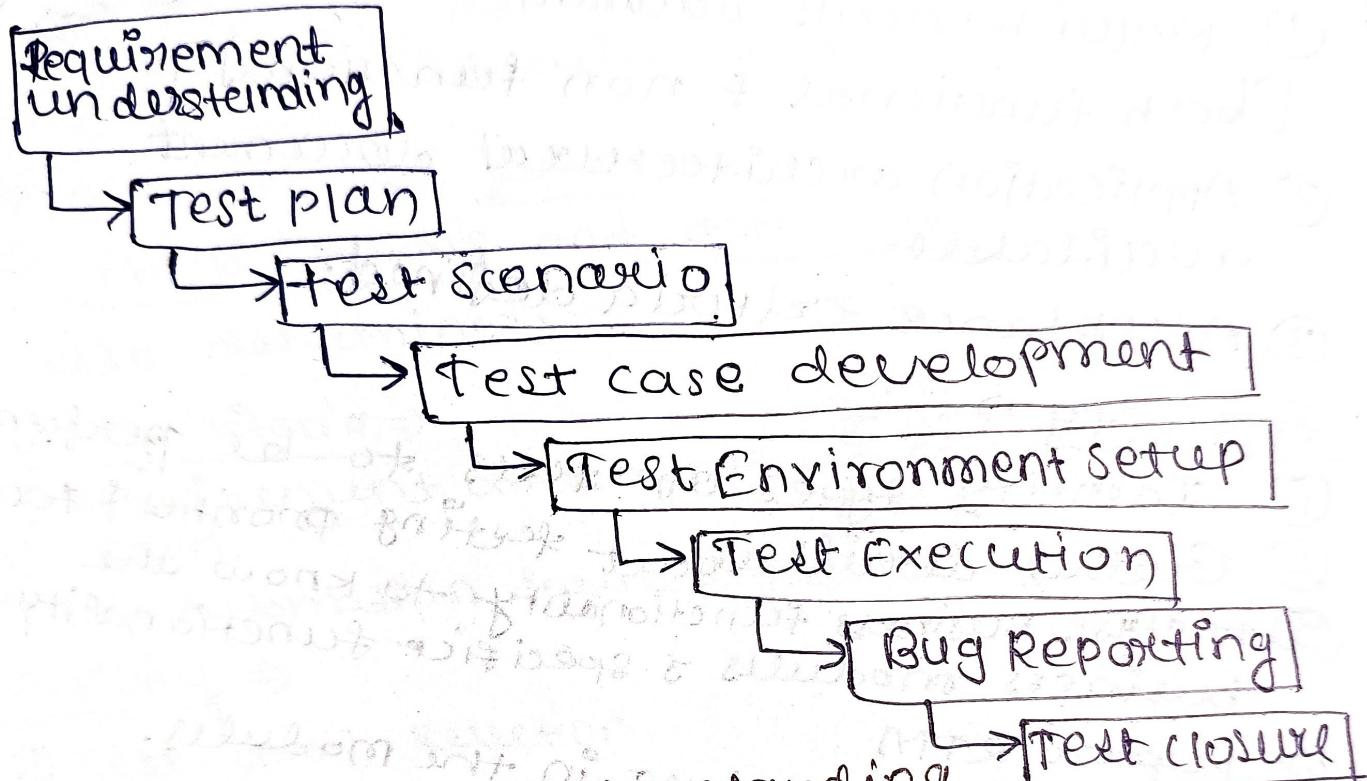
each of these stages has a definite entry & exit criteria, activities, deliverables associated with it.

Entry criteria ⇒ It gives the pre requisite items that must be completed before testing can begin. It is a set of condn that permits a task to perform.

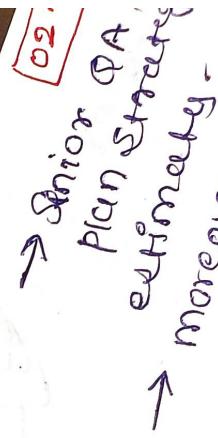
Exit Criteria ⇒ Exit criteria defines the items that must be completed before testing can be concluded. It is a set of expectations.

item / document / action / tasks

STLC Phases



- ① Requirement understanding
→ here, test team studies requirements from a testing point of view to identify testable requirements.



Entry criteria \Rightarrow

- ① Requirements document available [both functional & non functional]
- ② Application architectural document available.
- ③ Acceptance criteria defined.

Activity \Rightarrow

- ① Identify types of tests to be performed
- ② Gather details about testing priorities & focus.
- ③ Analyse business functionality to know the business modules & specific functionality.
- ④ prepare RTM
- ⑤ Identify transactions in the modules.
- ⑥ Identify all the user profiles.
- ⑦ Identify all the user interface / authentication, gather user interface / authentication, geographic spread requirements.
- ⑧ Identify test environment details where testing is supposed to be carried out.

Carried out automation feasibility analysis (if reqd)

Exit criteria \Rightarrow

- ① Signed off RTM
- ② Test automation feasibility report signed off by the client.

Deliverables \Rightarrow

- ① RTM
- ② Automation feasibility report (if applicable)

02. Test plan

- Senior QA manager determines the test plan strategy with efforts and cost estimate.
- moreover, the resources, test environment, test limitations and testing schedule are also determined.

Entry Criteria ⇒

- ① Requirement documents
- ② RTM.
- ③ Test Automation feasibility document.

- ① Approved PMP
- ② Approved SRS
- ③ Test plan guidelines
- ④ Test plan template

Activity ⇒

- ① Test tool selection.
- ② Test effort estimation.
- ③ Finalize on the best suited approach.
- ④ Analyze various testing approach available.
- ⑤ Preparation of test plan / strategy document for various types of testing.
- ⑥ Resource planning and determining roles & responsibilities.

Exit criteria ⇒

- ① Approved test plan / strategy document.
- ② Effort estimation document signed off.

test plan
should be
executed
& approved

Deliverables ⇒

- ① Test Plan / Strategy document.
- ② Effort estimation document.

04 Test

It involves test cases
of test cases

03 Test Scenario

Test scenarios ⇒

Identify all possible areas to be tested of what to be tested.

Entry criteria to identify Test Scenarios

① Approved test plan

② Approved SRS

③ Test scenarios guidelines

④ Test scenario template.

Exit criteria for test scenarios ⇒

① Test scenarios should be reviewed & approved (mapping test scenarios with requirements)

Test Scenario template

Company logo	Project ID Project Name Identified by Date Identified	Reviewed by Date Reviewed Approved by Date Approved				
TS#	Req#	Main fun	Test scenario description	Test case name / test ID	Document ID / Reference	Comments

04 Test case development

- It involves execution, verification & rework of test cases and test scripts ~~after the~~
- Develop the test cases based on scope & criterias

Entry criteria ⇒

- ① Requirements documents
- ② RTM & test Plan
- ③ Automation Analysis report

Activity ⇒ ~~reviewing requirements, test design, automation script (if applicable)~~

- ① Create test cases, test design, Automation script (if applicable)
- ② Review & baseline test cases & script
- ③ Create test data

Exit criteria ⇒ ~~Reviewed and signed test cases / scripts~~

- ① Reviewed and signed test cases / scripts
- ② Reviewed & signed test data & validation

Deliverables ⇒

- ① Test cases / scripts
- ② Test data

What is test case ⇒

Q
④
Test Execution ⇒

1.05 Test Environment setup

Test Environment setup ⇒

- It decides the SW & HW condn under which a work product is tested.
- An integrated environment is ready to validate the product.
- The test team is required to do a readiness check (smoke testing) of the given environment.

Entry criteria ⇒

- ① System design and architecture documents are available.
- ② Environment set up plan is available.

Activity ⇒

- ① Understand the required architecture, environment set-up.
- ② Prepare HW & SW development requirement list.
- ③ Finalize connectivity requirements.
- ④ Prepare environment setup checklist.
- ⑤ Setup test environment & test data.
- ⑥ Perform smoke test on the build.
- ⑦ Accept/reject the build depending on smoke test result.

Exit criteria ⇒

- ① Environment setup is working as per the plan & checklist.
- ② Test data setup is complete.
- ③ Smoke test is successful.

Deliverables ⇒

- ① Completed environment ready with test data setup.
- ② Smoke test results.

06 Test Execution

Test Execution \Rightarrow ① Real time validation of product & finding bugs.

- \rightarrow It is carried out by testers in which testing of the SW build is done based on test plan & test cases prepared.
- \rightarrow The process consists of test script execution, test script maintenance and bug reporting.
- \rightarrow If bugs are reported then it is reverted back to development team for correction and retesting will be performed.

Entry Criteria \Rightarrow

- ① Baselined RTM, Test plan, Test case /script are available.
- ② Test environment is ready.
- ③ Test data set up is done.
- ④ Unit / Integration test report for the

Activity \Rightarrow

- 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 application
- ⑦ Track the defects to closure

Exit criteria =>

- ① All tests planned are executed.
- ② Defect logged & tracked to closure

Deliverables =>

- ① completed RTM with execution status
- ② test cases updated with results
- ③ Defect reports

Q7. Test cycle closure

Test cycle closure ⇒

→ This phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices and test results.

Entry criteria ⇒

- ① Testing has been completed.
- ② Test results are available
- ③ Defect logs are available

Activity ⇒

- ① Evaluate cycle completion criteria based on time, test coverage, cost & Quality, critical business objectives
- ② prepare test metrics based on the above parameters
- ③ Prepare test closure report.
- ④ Qualitative & quantitative reporting of quality of the whole product to the customer.
- ⑤ Test Result analysis to find out the defect distribution by type & severity.

④ M T of
the issues,
the applicatio

Manual Testing

- MT is a type of SW testing in which test cases are executed manually by a tester without using any automated tools.
- The purpose of MT is to identify the bugs, issues and defects in the SW application.
- The testing checks the quality of the system & delivers bug-free product to the customer.

How to Perform M.T

- ① Read & understand the SW project document-ation/guide.
- ② Draft Test cases that cover all the requirements mentioned in the documentation.
- ③ Review & baseline the test cases with team lead, client (as applicable).
- ④ Execute the test cases on the AUT.
- ⑤ Report Bugs.
- ⑥ Once bugs are fixed, again execute the failing test cases to verify they pass.

PROS/ADV :-

- ① It is less expensive.
- ② Human can observe, judge & also provide intuition in case of mental tests.
- ③ Visual components like text, layout, other components can easily be accessed by the tester, & UI issues can be detected.

- ④ MT of an application identifies most of the issues, including the look & feel issues of the application.

Disadvantage / Cons :-

- ① MT is time consuming.
- ② It is not easy to find size difference of color combination of GUI objects using a manual test.
- ③ Load testing and performance testing is impractical in the manual tests.
- ④ When there is large no. of tests, then running tests manually is a very time-consuming job.
- ⑤ Regression test cases performed using manual tests are time consuming.

Why test Automation

- Test Automation is a best way to inc. the effectiveness, test coverage, and execution speed in SW testing.
- It is difficult to test for multilingual sites manually.
- Test Automation increases the speed of test execution.
- Test Automation helps inc. test coverage.
- Automation helps inc. test coverage.
- MT can become boring and hence error-prone.
- Which test case to automate?

Test cases to be automated can be selected using the following criteria to increase the automation ROI.

- i) High Risk → Business Critical test cases
- ii) Test cases that are repeatedly executed
- iii) Test cases that are very tedious or difficult to perform manually.
- iv) Test cases which are time-consuming.

The following category of test cases are not suitable for automation:

- ① Test cases that are newly designed & not executed manually at least once
- ② Test cases for which the requirements are frequently changing
- ③ Test cases which are executed on an ad-hoc basis.

Automated Testing process

- ① Test Tool Selection
- ② Define scope of Automation
- ③ Planning , design and development
- ④ Test Execution
- ⑤ maintenance

Benefits of Automation Testing

- ① 70% faster than the MT
- ② wider test coverage of application features
- ③ Reliable in results
- ④ Ensure Consistency
- ⑤ Saves time & cost
- ⑥ Improves accuracy
- ⑦ Human Intervention is not required during execution
- ⑧ Increases Efficiency
- ⑨ Better speed in executing tests
- ⑩ Reusable test scripts
- ⑪ Test frequently & thoroughly
- ⑫ More cycle of execution can be achieved through automation
- ⑬ Early time to market

Test Automation is a SW testing technique that performs using special automated testing SW tools to execute a test case script.

Test automation is the best way to increase the effectiveness, test coverage and execution speed in SW testing.

pros/adv → ① improves accuracy.
→ ② saves test execution time & money.
→ ③ test script execution written once can be reused.

- ① It helps you to find more bugs compare to a human tester. uses special tools to run test cases
- ② It supports various applications
- ③ Testing coverage can be increased because of automation testing tool never forget to check even the smallest unit.
- ④ It can easily increase productivity because it provides fast and accurate testing result.
- ⑤ AT is conducted using SW tools, so it works without tiring and fatigue unlike humans in MT.
- ⑥ As most of the part of the testing process is automated, you can have a speedy & efficient process.

cons/disadv

- ① The tools to run automation testing can be expensive, which may inc. the cost of the testing project.
- ② Test maintenance is costly.
diff. betw MT & AT

MT	AT
<ul style="list-style-type: none">① Testers manually execute test cases② It takes up lots of time & <u>implies</u> <u>high cost</u>③ we can manually test any applicatn④ Exploratory testing is possible.⑤ prog. knowledge is not must in AT required must⑥ does not use framework⑦ It is suitable for Adhoc, exploratory & Usability testing. It should also be used where the AUT changes frequently	<ul style="list-style-type: none">① uses tools to execute test case② test cases are automated, it saves time & is③ It is beneficial <u>only</u> for stable systems.④ does not allow random testing.⑤ No need for programming in MT⑥ does not use frameworks⑦ AT is suited for regression testing, performance testing, load testing or highly repeatable functional test case.

- ⇒ diff. levels of testing
- ⇒ If start with unit testing of ends
on Acceptance testing.
- there are 4 levels of testing
- ① unit testing
 - ② integration testing
 - ③ system testing
 - ④ acceptance testing

② what is test case?

- ① step by set of step-by-step instructions to verify that the system meets the requirements & works properly
- contains ⇒ title, description, test steps, expected result, actual result

what is exploratory testing?

- we need to explore the requirements of the project & identify all possible requirement.
- when there is no requirement at all then this testing is done.
- or even requirement is missing.

disadvantages you might have misinterpreted some features as being -

- ② more time consuming.
- ③ if any feature is missing then never come to know about that.

How to do

- ① Always try to understand the requirements of user to senior or manager about project
- ② user will doubt to seniors if understood the requirements fully to him.

Test Qualities

- ① No compromise on the qualities
 - ① Negative testing
 - ① Checking valid credentials (pre testing)
 - ② Checking invalid credentials (pre testing)
 - ② Analytical skills (thinking)
- ④ Patience
- ⑤ Communication & writing skills.

① 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.

- The main objective is to verify that the critical functionalities of the system are working fine. e.g. → New regi button is added in the login window & build is deployed with new code. we perform smoke testing on new build.
- A typical smoke test would be to verify that the application launches successfully, verify that the GUI is responsive, etc.
- ② Check that the GUI is responsive, etc.

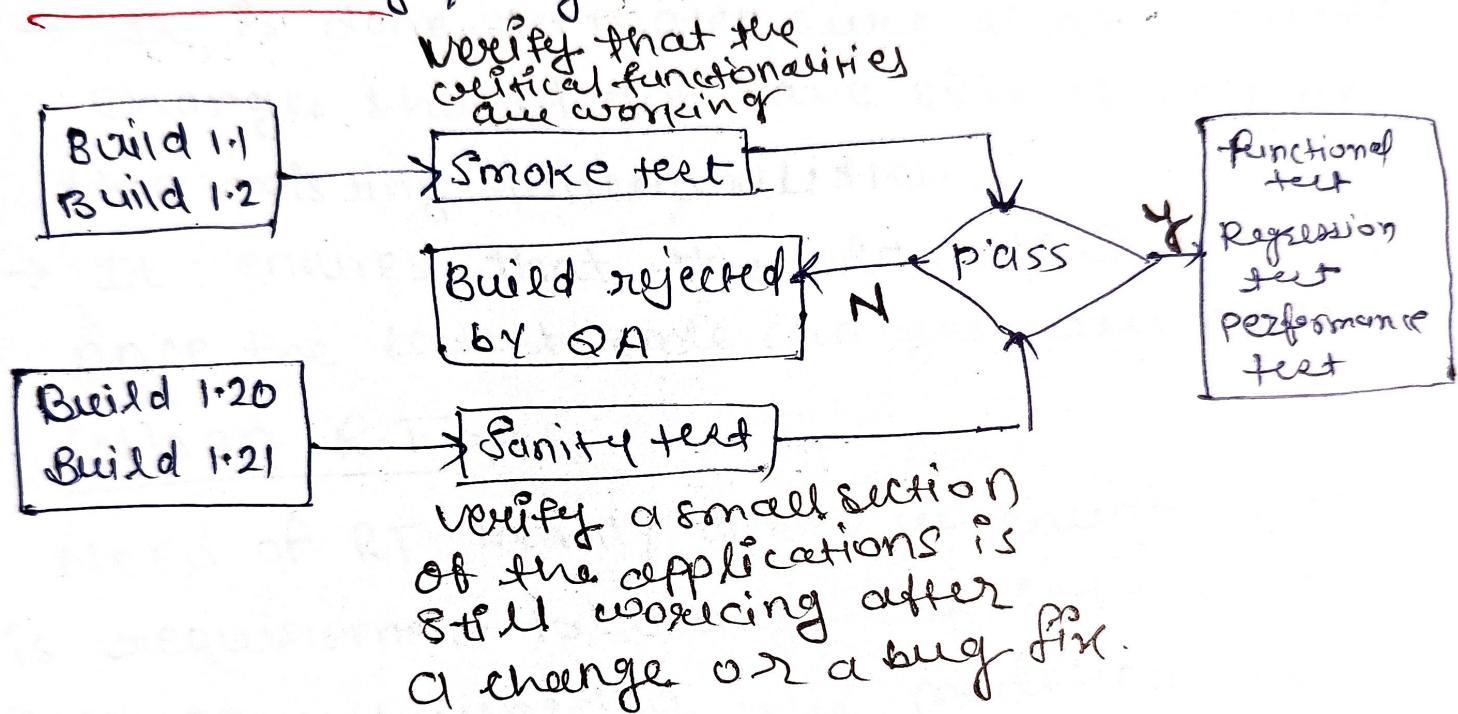
- It has a goal to verify "stability"
- It is done by developer & testers
- It verifies the critical functionalities of the system.
- It is documented / scripted
- It verifies the entire system from end to end
- It is a subset of acceptance testing.
- It is performed after receiving a new build.

build ⇒ executable file.

- we are going to testing
- high level functionality
- Build is releasing from development to tester

② Sanity Testing - executing high level tests

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

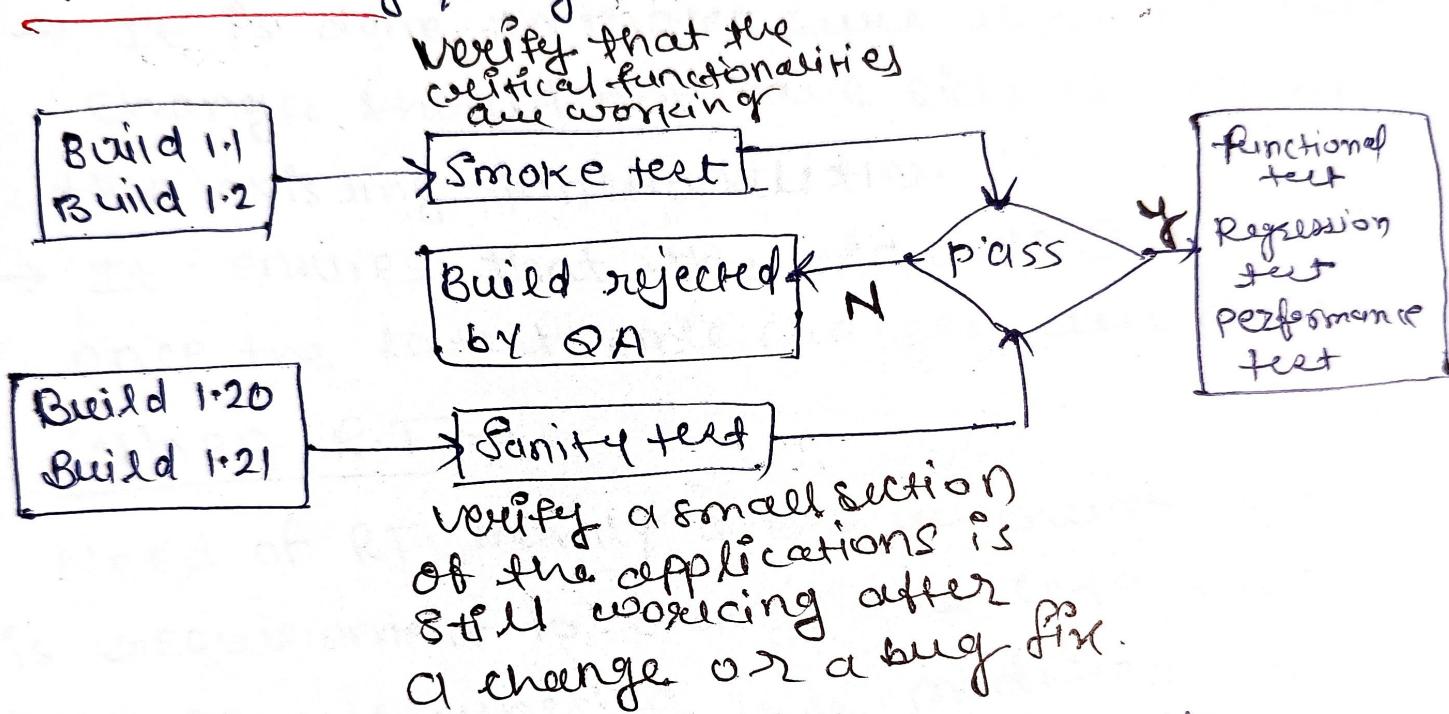


- It has goal to verify functionality
- It is done by testers
- It verifies new functionality like bug fixes.
- It is subset of Regression testing
- It is not documented / scripted
- verifies only a particular component.

Build is releasing from testing team to client
(not in environment)
combined all builds & releasing to the client

② Sanity Testing - executing high level test cases

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



- It has goal to verify functionality
- It is done by testers
- It verifies new functionality like bug fixes.
- It is subset of Regression testing
- It is not documented / scripted
- verifies only a particular component.

Build is releasing from testing team to client
(not in enrollment)
combined all builds & releasing to the client

Regression testing

- To check existing functionality is un affected whenever the new change is added.
- It is done to make sure that new code changes should not have side effects on the existing functionalities.
- It ensures that the old code still works once the latest code changes are done.

When RT ⇒

Need of RT mainly arises whenever there is requirement to change the code & we need to test whether the modified code affects the other part of the application or not.

RT is needed, when a new feature is added to the SW application & for defect fixing as well as performance issue fixing.

because of that bug fixed or adding new functionality is that existing func'tn affecting or not.

Re-testing \Rightarrow Testing defects were fixed or not in the current build

\rightarrow It means re-testing the functionality or bug again to ensure the code is fixed. If it is not fixed, defect needs to be re-opened. If fixed, defect is closed.

Check bug is fixed or not

Functional testing \Rightarrow It is a process of testing functionalities of the system & ensures that the system is working as per the specified document. The goal of the testing is to check whether the system is performing

F.T is a type of SW testing that validates the SW system against the functional requirements / specifications.

The purpose of functional test is to test each funcn of the SW applictn, by providing appropriate I/P, verifying the O/P against the functional requirements.

This testing checks the User Interface, API's, Database, Security, Client/Server Communication & other functionality of the application under test.

functional 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 =>

It 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 =>

It is used to determine if your SW is compatible with other SW elements of a system. with which it should operate, eg. Browsers, OS or HW. checks if the system runs well on diff. HW, OS, enviro & mobile devices

Database testing => user entered Data Database connection & user entered Data in application is saving into respective database tables.

Interface testing ⇒

Three areas to be tested here are -
application, web & Database Server.

Application ⇒

Test requests are sent correctly to the DB
& O/P at the client side is displayed correctly

web server ⇒

Test web server is handling all application
requests without any service denial.

Database Server ⇒

make sure queries sent to the db give
expected results.

Performance testing ⇒

testing page, data, images load time

Security testing ⇒ Security testing involves
the test to identify any flaws & gaps
from a security point of view.

Non functional testing ⇒

It is designed to test the readiness of a
system as per non functional parameters which
are never addressed by functional testing.

e.g. → non functional test would be to check how
many people can simultaneously login into
the system.

or tested here are -

Static testing

Testing an application without performing any action.

e.g. GUI testing, colors, spelling, alignment etc.

Identifies the defects in the early stages of development.

Dynamic Testing :-

Testing an application by performing required actions.

e.g. functionality testing, textbox, button etc.
validate the O/P with expected outcome

Ad-hoc Testing (monkey testing)

Testing the application without any proper planning. It is not structured.

e.g. if you are having 100 T.C. in a set of 100
randomly you perform some random test.
we prefer this testing, we don't have
enough time

for fixed testing as it is
a hard task to cover all the
possible errors by testing
program because of large number of
parameters

Advantages
1. It is a quick way to find bugs
2. It is a good way to find bugs
3. It is a good way to find bugs

Alpha testing => end to end testing, done by tester.

final testing on the application doing with in the development company

Beta testing => done by client

Testing doing in customer environment, This testing will be done by the customer or third party test engineers.

How to do functional testing =>

- ① understand the functional requirements
- ② Identify test I/P or test data based on requirements.
- ③ Compute the expected outcomes with selected test I/P values
- ④ Execute test cases
- ⑤ Compare actual & computed expected results.

Load system => measures system's behaviour under expected load.

Performance testing => checks if the application meets speed, scalability & stability requirement
stress => load is inc on the server till it crashes
Scalability =>

Test case development ① ③

- Test case \Rightarrow It is a set of actions executed to verify a particular feature or functionality of your SW application.
- A test case contains test steps, test data, precondⁿ, postcondⁿ developed for specific test scenario to verify any requirement.
- The test case includes specific variables or condn, using which a testing engineer can compare expected & actual results to determine whether a SW product is functioning as per the requirements of the customer.
- e.g. 1) Test scenario \rightarrow check login fun.

then test cases can be \rightarrow

TS1: Check results on entering valid user ID & password

TS2: check results on entering invalid user ID & password.

Test Case Template

Test Case template

business
company
log

Project History	Test Case History	Test Info
Project ID	Created By	Date Created
Project Name	Reviewed by	Date Reviewed
	Approved by	Date Approved
		Test Case ID:
		Date Executed
	Test Executed by	Version
		Build

Test Case
Test Case ID:
Test Case Name:
Test Case Status:
Test Case Result:
Test Case Defects:
Test Case Comments:
Test Case Reported:

TC#	TS#	Test design & steps	SIP date	Expected result	Actual Result	Pass	Fail	Not Executed	Comments	Defects
TC1	TS1	Test Step 1: Open Home Page	2023-09-01	Home page displayed correctly	Home page displayed correctly	Pass	0	0	0	0

Requirement Traceability Matrix

- 1) It is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled.
- 2) It is also known as Traceability matrix or cross Reference matrix.
- 3) It is also known as Traceability matrix whenever necessary to update the RTM whenever there is a change in requirement.
 - It captures all requirements proposed by the client.
 - main purpose of RTM is to see that all test cases are covered so that no functionality should miss while doing slow testing.
 - parameters include
 - Requirement ID (Business/technical)
 - Test case ID

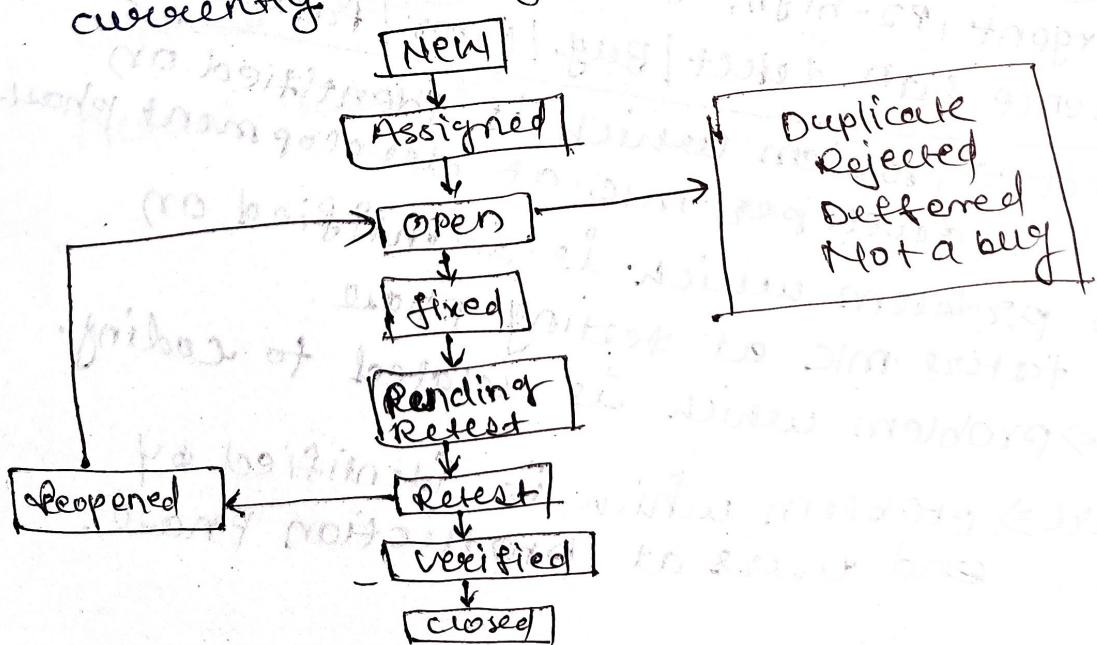
releasing

Bug Report

① Defect life cycle ⇒

- Defect life cycle in SW testing is the specific set of 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 assignees & make the defect fixing process systematic & efficient.

Defect states ⇒ It is in defect life cycle is the present state from which the defect or bug is currently undergoing.



Defect Management

defect status → New, open, fixed, closed,
Re-open, Not a bug, duplicate, need more info,
Can't reproduce.

Severity → Importance of defect w.r.t.
functional point of view

means criticalness of defect w.r.t. application

Severity classification could be →

Critical, high, medium, low

Priority → Importance of defect w.r.t. to client
point of view --- means how it should
be fix.

Priority classification could be →

P1 → urgent, P2 - high, P3 - medium, P4 - low

Difference b/w defect / Bug / Error / failure ⇒

- ① Defect ⇒ problem which is identified on developer's side at development phase.
- ② Bug ⇒ problem which is identified on tester's side at testing phase.
- ③ Error ⇒ problem which is related to coding.
- ④ 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:

RTM =>

- ① RTM captures all requirements proposed by the client.
- ② In other words, it is a document that maps & traces user requirement with test cases.
- ③ The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while doing SQA testing.
A simple way is to trace the requirement with its corresponding test scenarios & test cases. This merely termed as RTM.
- ④ ~~RTM~~ → traceability matrix is typically a worksheet that contains the requirements with all possible test scenarios & cases & their current status, i.e. if they have been passed or failed.

RTM \Rightarrow

- ① RTM captures all requirements proposed by the client.
- ② In other words, it is a document that maps & traces user requirement with test cases.
- ③ The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while doing SQA testing.
A simple way is to trace the requirement with its corresponding test scenarios & test cases. This merely termed as RTM.
- ④ ~~RTM is~~ traceability matrix is typically a worksheet that contains the requirements with its all possible test scenarios & cases & their current state, ie if they have been passed or failed.

parameters →

- ① Requirement ID
- ② Requirement Type & description
- ③ Test cases with status

Req. No	Req Desc	Testcase ID	Status
123	Login to the system	TC01, TC02, TC03	TC01 - pass TC02 - pass

details template

sno	Req_ID	Reqdesc	TC2D	SC Desc	test design	test design

UAT test Req?	Test Execution	defects	defected
	Test Env UAT Env Prod Env		

defect status	Req. coverage status
---------------	----------------------

Types of traceability test

- ① Forward T
- ② Backward T
- ③ Bi-directional T (F+B)

How to create RTM →

→ on the basis of BRD & TRD

BRD → business requirement document

TRD → Technical / functional

Business Requirement #	Technical Requirement #	Test case ID
B1	T94	1
B2	T95	2

~~"evid." entering "my"~~

ADV → ① It confirms 100% test coverage

② It highlights any requirements missing or document inconsistencies

③ It shows the overall defects or execution status with a focus on business requirements.

④ It helps in analyzing estimating the impact on the QA teams while w.r.t. revisiting or reviewing on the test cases

~~Test.~~

Positive testing ⇒

It is the process of inputting valid data & making sure the application responds as expected.
Exceptions (errors) are not expected in positive testing.

Negative testing ⇒

It is process of inputting invalid data & making sure the application responds as expected. Exceptions (errors) are expected in negative testing.

Exploratory testing ⇒

It is a type of SW testing where test cases are not created in advance but testers check system on the fly. It is widely used in Agile models & is all about "discovery", "investigation" & "learning".

~~want enter~~

What are cookies?

cookies are files which are stored on your computer, smartphone, tablet or other device when you browse the internet.

They are designed to hold data specific to a particular user & website, & can be accessed either by a web server or the user's computer or device.

Test case Design techniques

④

① Each SW testing techniques help you design better test cases.

② Boundary Value Analysis :- divide a set of test condn into groups

→ BVA is based on testing at the boundaries
betn partitions.
It includes maximum, minimum, inside or outside boundaries, typical values & error values.

e.g. PIP condn is valid betn 1 to 10
Boundary values 0, 1, 2 & 9, 10, 11

a to b → a-1, a, a+1, b-1, b, b+1

③ Equivalence class partitioning :-

It allows you to divide set of test condn into a partition which should be considered the same. This SW testing method divides the IIP domain of a program into classes of data from which test cases should be designed.

e.g. IIP condn are valid betn 1 to 10 & 20 to 30
→ to 0 (invalid) you will divide your range

1 to 10 (valid) into various classes & from each

11 to 19 (invalid) class you will test for a single value

20 to 30 (valid) if it is passed whole class is

31 to -- (invalid) if it is failed whole class is failed

③ Decision table Based testing

- It is also known as cause-effect table.
- This SW testing technique is used for functions which respond to a combination of IIP or events.
- The first task is to identify functionalities where the OIP depends on a combination of IIP.
- If there are large IIP set of combinations, then divide it into smaller subsets which are helpful for managing a decision table.
- following all steps to create a decision table
 - ① Enlist the IIP in rows.
 - ② Enter all the rules in the col^m
 - ③ fill the table with the diff. combinatⁿ of IIP
 - ④ In the last row, note down the OIP against the IIP combination.

e.g A submit button in a contact form is enabled only when all the fields are entered by the end user

	Rule	2	3	4	5	6	7	8
SIP	F	F	B	B	F	F	F	F
Name	F	F	T	T	F	F	T	T
Email	F	T	F	T	F	T	F	T
msg	F	F	F	F	T	T	T	T
OIP								
Submit	f	f	F	F	F	F	F	T

(2) State Transition

We test if the changes in input condⁿ change the state of the AUT. This testing technique allows the tester to test the behaviour of an AUT.

① ECP \Rightarrow

divide a set of test condⁿ into groups which is having similar behaviors to reduce no. of test cases.

e.g.

Age (accepts 1 to 60)

Invalid	Valid	Invalid
0	1-60	61

② BVA \Rightarrow

BVA is based on testing the boundaries of condⁿ formula \Rightarrow min, max, min-1, max+1

Valid boundaries \Rightarrow min, max

Invalid boundaries \Rightarrow min-1, max+1

e.g.

Name (accepts 5 to 10 characters)

Invalid (min-1)	Valid (min, max)	Invalid (max+1)
4	5, 10	11

③ State transition table

Application provides diff. O/P for same IP
~~base~~ based on previous stage

→ provide same IP for multiple times \Rightarrow diff. technique
 needed to get O/P as diff.

eg.

	Username	Password	Result
Username			
Correct	correct	correct	valid password
Correct	wrong	wrong	invalid password
Correct	wrong	wrong	invalid password
Correct	wrong	wrong	Account locked
Username			Reset
Correct	correct	correct	Login successful
Correct	w	w	invalid pass
w	w	c	—
w	w	w	—

Decision table → testing with diff. combination of IP which produce different results

① what is SW?

→ It is set of instructions, data or programs used to operate computers & execute specific task.

② what is SW testing?

- Identify defects in SW
- test developed application, whether it is working according to client requirement or not.
- To ensure that SW product is defect free.
- it also involves implementation of product modules
- identify errors, gaps & requirements of an element in project or system.

Q.W.Que

① Different methods of SW testing

There are the 3 most imp. techniques is SW testing like Black Box testing, White Box testing and Grey Box testing

- Black \Rightarrow ① also called Behavioural testing
Box ② Based only on requirements & specifications
③ No consideration of internal code structures, implementation details & internal paths of SW

- White box \rightarrow ① also called clear box, Glass box, structural or code based
② Validate internal structure of working of code
③ needs programming skills to understand internal code structure

Grey \rightarrow ① combination of white & black box

- ② some knowledge of internal structure required.

Responsibilities

- i) Understanding the Requirements & functional specification of the project.
- ii) Identified Test Scenarios required for testing.
- iii) participated in designing the Test cases.
- iv) prepared & Executed Test Cases as per System Requirements.
- v) Defect Reporting and Tracking using JIRA.

Test Suite creation

i) worked using Page object model with Junit, Testing & data driven for UI auto

ii) selenium was used to generate automated test scripts for functionality.

* Extensively performed MT process to ensure the quality of the SW.

- Then An Automation tester is a person who writes the scripts to test the correctness of the applicatn.
- Test Automation involves creating & applying technologies that control or monitor production and timely delivery of software products. Automated testing helps in reducing the time consumed to perform tedious tasks.
- The main objective of a test automation engineer is to automate as much of the testing as possible with a minimum set of code or scripts.
- An automation tester designs the test cases by creating scripts that check the functionalities automatically.

- ① what is diff betⁿ project & products?
- project is developed based upon Customer requirement.
 - whereas product is developed based upon market requirement.

Basic
que d

SOLC

① It is P
cycle
overall
SICL de

② diff b
project

produ

prep

③ wr

tr

etc

④ di

QA

QC →
etc

Real time QM que (Inr video by Sdet)

Basic
Que → diff betn SDLC & STLC

SDLC	STLC
① Pt is simpler life cycle overall an process of SICL dev process	② QA is testing process in SDLC
③ diff betn project & product <u>project</u> → It is developed based on customer requirement.	
<u>product</u> → Pt is developed based on market requirement, user requirement. we need to first capture user requirement then need to prepare the product.	

③ what is V-model?

→ model is in V model ; development & testing will do parallelly.

④ diff betn QA & QC ?

QA → It is talk about complete process
↳ manager, preventing, entire SDLC process

QC → It is talk about testing activity

↳ testers, detection, testing process

?
test scenarios?
?
test cases?
? test scenarios?



- ⑤ unit testing →
testing a single program or component of project. done by developer.
- ⑥ Integration testing →
testing an one combination of one or more modules is called integration testing
- ⑦ System testing →
Testing full functionality of system.
Testing end to end functionality of application w.r.t. customer requirement.
- ⑧ types of System testing →
- ① functional
ui, flows, navigation, db, business logic, elements displaying or not
 - ② Non-functional
Once functionality is stable then go for non functional performance, load, stress, security, recovery
- ⑨ diff. betn whitebox & blackbox
- whitebox
Test internal logic of program by developer
- blackbox
Test the functionality, w.r.t. to customer requirement.
- ⑩ verification → done before SW developed
It is static technique
document testing.
Correctness & completeness of document is imp.
validate documentation
- Validation
→ It comes in testing part.
System testing, UAT, dynamic testing
→ done after SW developed

⑪ GUI
Graphical
cg. elements
all com
we need
element
cg. fo
color &
or not
not

⑫ DIP

⑬ EC
✓ valid
✗ invalid
valid
a - z

⑭ C

✓ is
use

AT? test scenarios?
oratory testing?

GUI Testing → look & feel of apps as per client
Graphical User Interface Testing

e.g. elements, radio button, checkboxes, links this all comes under GUI.

We need to check the behaviour of this element.

e.g. font, Spelling mistake, alignment, font color & size, all being properly displayed or not, all links properly aligned or not.

⑫ DB domain testing & its test techniques

Verify the input boxes w.r.t requirement

① ECP ② BVA → talk about range

✓ validate boundaries

invalid data

Valid	Invalid
a - z	A - Z
0 - 9	

⑬ What is db testing?

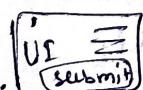
Testing DML operations w.r.t. to db

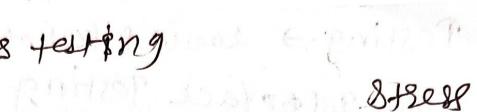
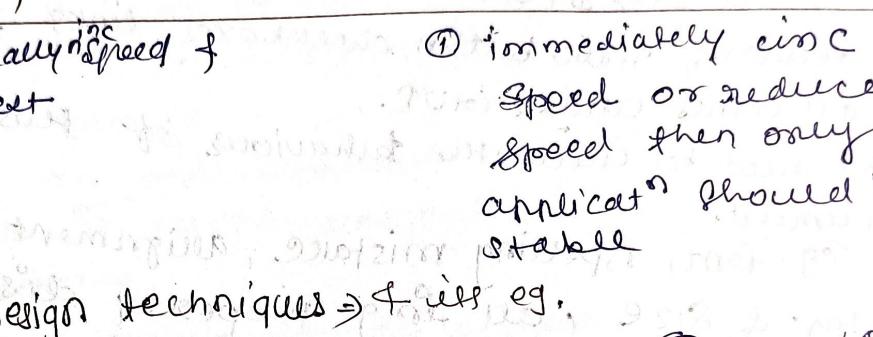
① Sometimes view or connection problem

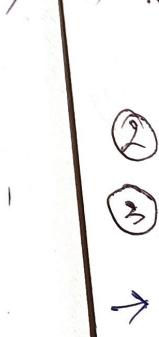
data is not stored in db

② So, we need to verify the frontend in frontend affecting in db or not

③ Testing DML → inserting data updating data deleting data selecting



- 
- ⑭ Load & stress testing → 
- Load Stress
- ① gradually increase speed & then feet ① immediately circ. Speed or reduce speed, then only applicatⁿ should be stable
- Sanity →
- ⑮ Test design techniques → 
- ① BVA ② ECP ③ decision table ④ transitⁿ diag
⑤ error guessing
- ⑯ Give eg. 
- ⑰ ad hoc testing →
→ randomly test the applicatⁿ by knowing the functionality.
→ perform when after completing all testing have time then perform stress testing to find corner scenario.
- ⑱ exploratory testing?
don't have knowledge on applicatⁿ, but go through as we are exploring knowledge on applicatⁿ by working with an applicatⁿ, just randomly go through all lines, all functionality.
→ when don't have documentedⁿ, test cases ready is ready but test don't have test cases

- ⑲ end of com → PROC
⑳ 
- ㉑ 
key
- ㉒ 
+ d
①
- ㉓ 
②
③
→

story telling
a test scene

- (19) smoke → initial level of testing
e.g. page navigation

① installers part (Build is properly installed or not)
② basic

Sanity → ① verify basic ~~features~~, functionality

(20) end to end testing → complete functionality of application we have to test.
complete application w.r.t. to customer process related all

(21) use case ⇒
use case is a requirement described by the product owner or BA.

(22) test case ⇒
describe how to test:

test case → test scenario
how to test → what to test

① what is the testing process followed in your company?
Agile Testing process.

② Explain process in your company?

③ flow to report bug → review the bug 1st
→ verify that bug in your environment 1st
→ then I will set severity & priority to that bug
& then reporting bug to developer by providing

- ⑭ 1) and 2) Screenshot, reproducible steps all other related info.
- ⑮ what is defect life cycle?
- ⑯ what is priority & severity
- importance of defect seriousness of defect
How early defect of own how defect impact
should fix) on system
- ⑰ Give me an eg. High severity & low priority
high priority low severity
- ⑱ how to write test cases? what are TPLP
as soon as get requirement / story / epic / sprint
Once it is reviewed then start writing story
test cases for particular story.
TIP for writing test case → use screen, use real
once write test cases then we need to
reviewed with team & then finally
approved with product owner.
- ⑲ what is your role / responsibilities / day to day activities?

AT
test scenarios
& test cases
rotatory testing



all other

FAQ on project

① Explain your project

→ explain Domain → ecom, healthcare
portal, banking, insurance
financial

→ for which client you have developed this
project

→ diff. technology used in project

→ tools you have used (team size)

→ How many people you work

→ How many people you work like

→ tell about project like

functionality, UI, outcome of project

cenat are diff. functionality you

tested

How you tested?

What are the test cases you have
written?

② How many test cases have you written
for your project?

depends on size of project

Small project (3-months) → 10 TC / day • $150 \text{ TC} / \text{Month} = \frac{150}{3}$

medium (6 months) → approx $150 \times 6 = 900 \text{ TC}$

large (1+ year)

Small = 20-25 TC
Large = 5 TC

job related
Apps → naukri.com, indeed, internshala,
freehanded, linkedin, job news 7

gen → Hindi dict, mgjio, adobe reader (Airtel),
bankers point, can scanner, english
Hello eng, oALD, eng Spoken guru, Vi,
video show, zoom, WPS office

bankers pt →

All → How many defect you reported in your
project?
varied from project to project

Number

3-month (#5 - 20)

All → You found a defect in QA, but dev
is not able to reproduce it, unless you tell dev
it's not a defect

All → You reported a defect, dev says it's
not a defect it is as per require

approach product owner

All → what are diff defect metrics
follow

AT ?
factory testing?
test scenarios?

Phone imp info

SSD drive → for improve PC performance

fb pass → iamhappy / iamhappysomuch
iamsomuchhappy

LinkedIn crme →

xerox pointout ⇒ Jayxerox666@gmail.com.

rohinihirechan10 ⇒ pass ⇒ Rohini@9096

9 ⇒ 9096676190

8 ⇒ -11

2020 ⇒ :11

rohinihirechan2020@gmail.com ⇒ .c-11-

email ⇒ rohinihirechan2020@gmail.com

pass ⇒ Rohini@7276

[Skype] ⇒ username ⇒ 9096676190

pass ⇒ Rohini@123

[Wordpress] ⇒ Username ⇒ Rohini

password ⇒ Rohini@123

[MongoDB] ⇒ rohini2020 ⇒ pass ⇒ Rohini@123

fresher's world

- ① Educational Details
- ② Personal details
- ③ Career related info.
- ④ career preferences

Github → rohini-hirechand@gmail.com
Pass → Rohini@9096

LinkedIn → email → 9096

Pass → Daizy@123

Rohini@9096

16071996

Hackerrank → email → 2020 → Rohini@123

HackerEarth → 2020 → Rohini@123

TopCoder.info → Username = Rohini_Hirechand

Pass = Rohini@123

Hackerathon → 2020 → Pass → Rohini@gmail.com

Mom to pass → Rohini@123

Fresherworld.com → rohini-hirechand@gmail.com → through
Pass → Rohini@123 → Google account

TimesJobs → rohini-hirechand@gmail.com
Rohini@123

factory testing?
AT? test scenarios?

Imp Info

internship at qo id

Indeed \Rightarrow profile \Rightarrow go id

\rightarrow Q what is your most recent job title?

- \rightarrow about you
- \rightarrow work experience
- \rightarrow education
- \rightarrow skills / IT skills
- \rightarrow certifications

Additional \Rightarrow Q Language

② online profile \rightarrow address of your blog,
personal homepage, fb or twitter prof.

LinkedIn

③ Awards & Achievements

④ NGO / Student Organization

⑤ patents

⑥ projects / papers presented.

Device key \Rightarrow OM64612E 0D1991F87
OAE