☑ Full stack QA :-

i) Manual testing (Without using any tools)
ii) Automation → a) Java Programming
b) Selenium.
iii) Advance Automation → a) Maven
b) Jenkins
c) Git
d) Github
e) Docker
iv) Manual testing Project
v) Automation testing Project (Using tools)
vi) FAQs
vii) Materials / Assignments.

Manual Testing

Module 1 : Testing Concept (Theory)
(What to test)

Module 2 : Testing Project (Practical)
(How to test (Techniques))

Module 3 : Agile Process (Jira (Agile tool))

xBank (Company) → IT Company → Develop → Test → Deliver → xBank

Introduction

1) What is software?

→ A software is a collection of computer programs that helps us to perform a task.

2) Types of software?

→ a) System Software :- Ex → Device Drivers, Operating Systems, Servers, Utilities etc.

b) Programming Software :- Ex → Compilers, Debuggers, Interpreters etc.

c) Application Software :- Ex → Web Applications, Mobile Apps, Desktop Applications etc.

3) What is Software Testing?

→ i) Software testing is a part of software development process.

ii) Software testing is an activity to detect and identify the defects in the software.

iii) The objective of testing is to release quality product to the client.

4) What is Software Quality?

→ Quality :- Quality is defined as justification of all the requirements of a customer in a product. [Note : Quality is not defined in the product. It is defined in the Customer's mind].

Quality Software is reasonably
i) Bug free
ii) Delivered on time
iii) Within budget
iv) Meets requirem or expectations
v) Maintainable

5) Product vs Project :-

→ If software application is developed for specific customer based on the requirement then it is called Projec

→ If software application is developed for multiple customers based on the market requirements then it is called Product.

6) Why do we need testing ?

→ i) Ensure that software is bug free
ii) Ensure that system meets customer requirements and software applications.
iii) Ensure that system meets end user expectations.
iv) Fixing the bugs identified after released is more expensive.

7) Error, Bug and Failure :-

→ i) Error → Any incorrect human action that produces a problem in the system is called an error.
ii) Defect / Bug → Deviation from the expected behavior to the actual behavior of the system is called Defect / Bug.
iii) Failure → The deviation identified by end-users while using the system is called a failure

8) Why the software has bugs normally?

→ i) Miscommunication or no communication
ii) Software complexity
iii) Programming errors
iv) Changing requirements
v) Lack of skilled testers

9) SDLC (Software Development Life Cycle) :-

SDLC is a process used by software industry to design, develop and test softwares.

In any company 3 Ps are important pillars :-

** i) People
ii) Process
iii) Product

☐ SDLC Process :-

i) Requirements Analysis (Project Manager)
ii) Design (Designer)
iii) Development (Coders/Developers)
iv) Testing (Testers/QA/QC)
v) Maintenance.

10) Waterfall Model :- (Documentation)*

Old and traditional model. (Linear model).

(Product/Project Manager collect the requirement from customer)

Requirement Analysis → System Design (Based on requirement Designer will design the model). → Implementation (Development) → Testing → Deployment → Maintenance

(Each phase is depend on previous phase).

☑ Advantages of Waterfall Model :-

i) Quality of the product will be good.

ii) Since requirement changes are not allowed, chances of finding bugs will be less.

iii) Initial investment is less since the testers are hired at the later stages.

iv) Preferred for small projects where requirements are freezed.

☑ Disadvantages of waterfall model:-

i) Requirement changes are not allowed.

ii) If there is defect in requirement that will be continued in later phases.

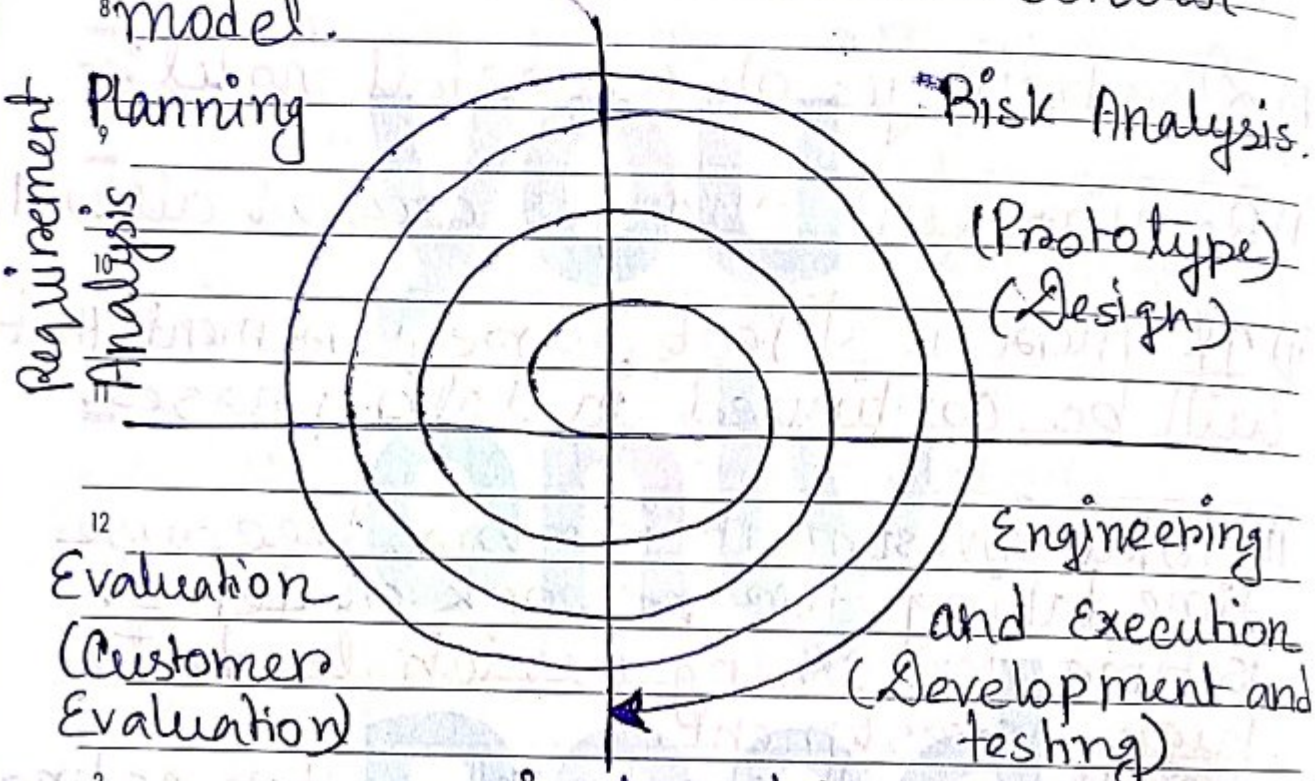
iii) Total investment is more because time taking for rework on defect is time consuming which leads to high investment.

iv) Testing will start only after coding.

II) Spiral Model (Another model of SDLC Process)
(Preferred for Product based Companies)

i) Spiral Model is iterative model.
ii) Spiral Model Overcome drawbacks of Waterfall model.
iii) We follow spiral model whenever there is dependency on the modules.
iv) In every cycle new software will be released to customer.

v) Software will be released in multiple versions so it is also called version control model.



Spiral model.

Advantages:-

i) Testing is done in every cycle, before going to the next cycle.

ii) Customers will get to use the software for every module.

iii) Requirement changes are allowed after every cycle before going to the next cycle.

Disadvantages :-

i) Requirement changes are not allowed in between the cycle.

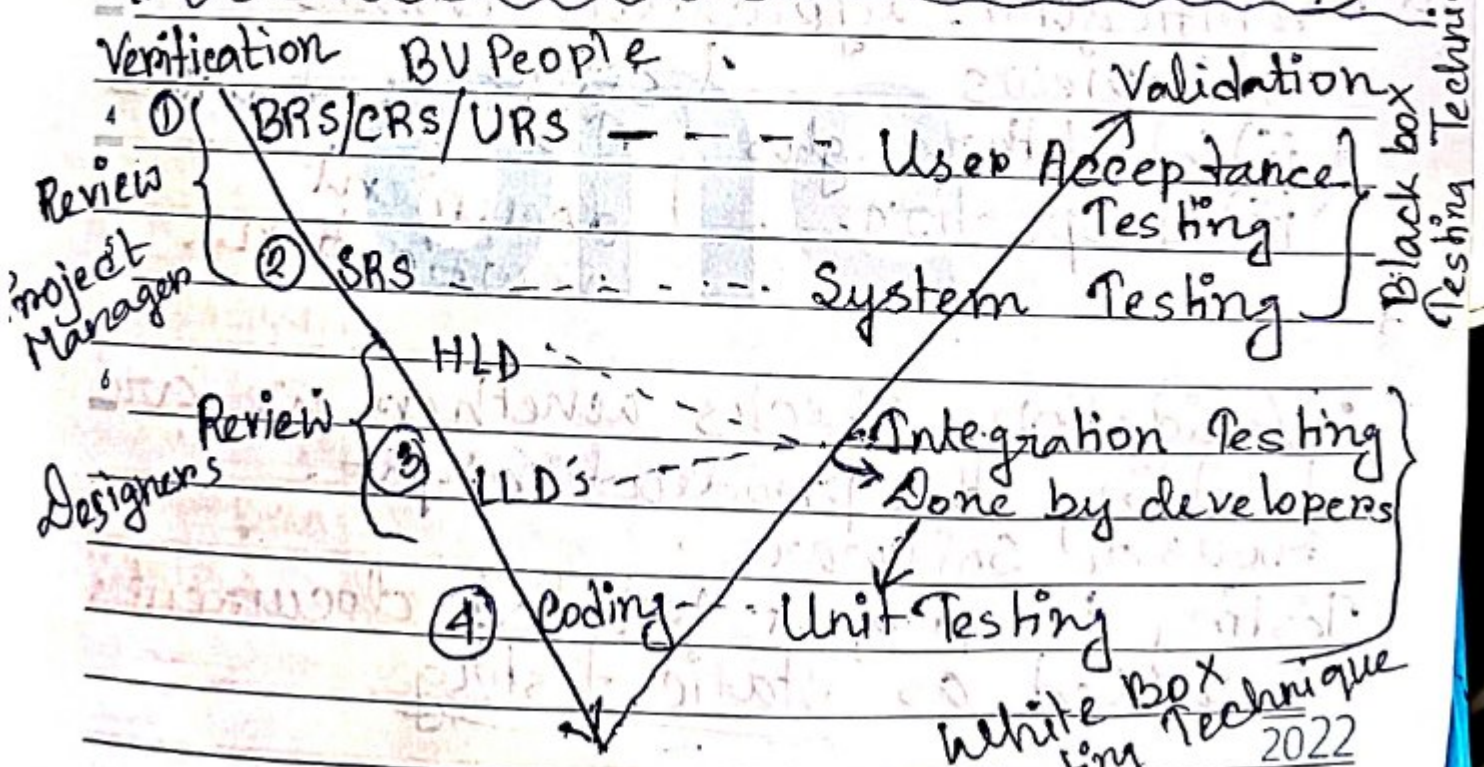
ii) Every cycle of spiral model looks like waterfall model.

iii) There is no testing in requirement and design phase.

Prototype :- Blueprint of the software.

Initial requirements from the customer → Prototype → Customer → design, coding, testing.

12) V-model / VV model :- (Verification and Validation model)



BRS → Business Requirement Specification.

CRS → Customer Requirement Specification

VRS → User's Requirement Specification

SRS → Software Requirement Specification

HLD → High level Design documents

LLDs → Low level Design documents.

13) Verification Vs Validation :-

- Verification checks whether we are building the right products.
- Focus on documentation

Use Static Testing.

- Verification typically involves →
  - i) Reviews
  - ii) Walkthroughs
  - iii) Inspection.

  } Static Testing techniques document verification.

- Validation checks whether we are building the product right.

Focus on Software.

Testing the project related documents is called as static testing.

In black box testing technique → No need of coding. (Done by Testers).

White box testing → Done by coder himself.

Testing the actual software is called as Dynamic Testing.

i) Unit testing
ii) Integration testing
iii) System Testing
iv) UAT (User Acceptance Testing)

Advantages of V model :-

i) Testing is involved in each and every phase.

Disadvantages :-

i) Initial investment is more.
ii) Documentation is more.

Static testing is an approach to test project documents in the form of review, walkthrough and inspection.

14) Static Testing :-

i) Review
ii) Walkthrough
iii) Inspection.

1) **Review** :- Conducts on documents to ensure correctness and completeness.

Example :- i) Requirement Reviews
ii) Design Reviews.
iii) Code Reviews
iv) Test plan reviews
v) Test cases reviews.

2) **Walkthrough** :- It is a ~~normal~~ informal review.
Author reads the documents or code and discuss with peers. It is not pre-planned and can be done whenever required. Also walkthrough does not have minutes of the meet.

3) **Inspection** :- Its a most formal review type. In which at least 3-8 people will ~~seat~~ sit in the meeting 1-reader 2-writer, 3-moderator (Organizer)

Inspection will have a proper schedule which will be intimated via email to the concerned developer / tester.

15) QA, QC, QE :-

QA → Quality Assurance
* QA is process related.

3 Pillars of Company. 1) People (QC) (People)
2) Process (QA)
3) Product

QC → Quality Control.
* QC is the actual testing of the software.

QA	QC
i) QA focuses on building in quality	i) QC focuses on testing for quality.
ii) QA is preventing defects	ii) QC is detecting defects.
iii) Process Oriented	iii) Product Oriented.
iv) QA for entire life cycle	iv) QC for testing part in SDLC.

QE → Quality Engineering (Automation Tester)
(Also write the code).

16) Levels of Testing :-

i) Unit Testing
ii) Integration Testing
iii) System Testing
iv) User Acceptance Testing (UAT)

→ White box testing Technique.

i) Unit Testing :-

- A unit is a single component or module of a software.
- Unit testing conducts on a single program.
- White box testing technique
- Conducted by developers.

Unit testing techniques are →
i) Basis path testing.
ii) Control structure testing
a) Conditional coverage
b) Loop coverage
iii) Mutation testing

2022 Ex → See the PDF (3)

II) Integration Testing :-

- Performed between two or more modules.
- Checks data communication between multiple modules.
- White box testing technique.

Ex:-

Module 1 → Module 2

[Login] → [Mail Box]

Mail Box ↓ Delete Action

Module 3 [Deleted Mails]

Two types :-

i) Incremental Integration Testing :-
Incrementally adding the module and testing the data flow between the modules.

2 Approaches
i) Top down
ii) Bottom up

Module 1
Module 2
Module 3

i) Top down :- Ensure the module add is the child of previous module.

[Compose Email] · Parent Module.

[Sent items] Child module

ii) Bottom up :- Ensure the module added is the parent of the previous module.

Compose Email — Parent module

Sent items — child module.

• Combination of Top-Down and Bottom up approach is called Sandwich Approach.

ii) Non-Incremental Integration Testing/ Big Bang Testing :-

• Adding all the modules in a single shot and test the data flow between modules.

Drawbacks :-

• We might miss data flow between some of the modules.

• If you find any defect we cant understand the root cause of defect.

iii) System Testing :- Testing over all functionality of the application with respective client requirements.

- Black box testing technique.
- Conducted by testing team.

System Testing focuses on →

i) User interface testing (GUI) → Graphical
ii) Functional Testing **
iii) Non Functional Testing
iv) Usability Testing.

iv) User Acceptance Testing (UAT) :-

UAT Teams conducts acceptance testing in 2 levels :- (Black box testing)

i) Alpha testing
ii) Beta testing.

i) GUI Testing :- System testing

Graphical User-Interface testing focuses on front-end part. It is a process of testing the user interface of an application.

It includes all the elements such as menus, checkbox, buttons, colors, fonts, sizes, icons, content and images

GUI Testing Checklist :-

- Testing the size, position, width, height of the elements.

• error messages that are getting displayed.
• Different sections of the screen.

ii) Usability Testing :- • During this testing validates application provided context sensitive help or not to the user.
• checks how easily the end users are able to understand and operate the application is called usability testing.

iii) Functional Testing :- Functionality is nothing but behaviour of application.
Functional Testing talks about how your feature should work.

* i) Object Properties Testing :- Check the properties of objects present on the application.
Ex → Enable, disable, visible, focus....

* ii) Database Testing :- / Backend testing :- Mainly focuses on Data Manipulation language (DML) operations.

Requirement from the customer

UI ⟷ Data

i) insert iii) delete } DML
ii) update iv) select

It is a grey box testing. (Both white box and black box testing).

Tester should know SQL, PLSQL

1) Table and column level validations (Column type, column length, number of colums..

2) Relation between the tables (Normalizatio

Functions Triggers Views
Procedures Indexes etc...

Database testing :- Checking database operations with respec to user operations.

* iii) Error Handling :- Tester verify the error messages while performing incorrect actions on the applications. Error messages should be readable. User understandable/simple language.

* iv) Calculations or Manipulations Testing :- Tester should check the calculations.

* v) Links Existence and Links Execution :-
Where exactly the links are placed. (Link existence).

links are navigating to proper page or not.
(Link execution)

3 types of links →
i) Internal links
ii) External links
iii) Broken links.

i) After clicking the link if it navigate to the same page but different section. (Internal)

ii) click the link, navigate to the another page. (External)

iii) click on the link which will not navigate to the other page (Broken)

*vi) Cookies and Session :-
Cookies are temporary files which are created by the browser while you browse the ~~data hero~~ page through internet.

Sessions will be created on the server site. Sessions are time slots created by the server. Sessions will be expired after some time (If you are idle for some time).

Expection of the customer

Expectation of the customer

iv) Non-Functional Testing :-

Once the application functionality is stable then we do Non-Functional testing.
Focus on performance, load it can take and security etc.

i) Performance Testing :-
Speed of the application after installation.
a) Load testing
b) Stress testing
c) Volume testing

Gradually → (customers)

a) Load :- Increasing the load on the application ~~slowly~~ and then check the speed of the application.

b) Stress :- Suddenly increase/decrease the load on the application and check speed of the application.

c) Volume :- check how much data is able to handle by the application.

ii) Security Testing :- How secure our application → users are valid or not
a) Authentication :-
b) Authorization / Access Control :- Permissions of the valid user.

iii) Recovery Testing :-
checks the system change to abnormal to normal.

iv) Compatibility Testing :-
a) Forward compatibility → Update software (new version)
b) Backward "
c) Hardware "
→ (Configuration testing)

v) Installation Testing :-
check screens are clear to understand, Simple or not, Un-installation process, Screens navigation.

vi) Sanitation / Garbage Testing :-
If any application provides extra features / functionality then we consider them as bug.

Functional	Non-functional
i) Validates functionality of software	i) Verify the performance, security, reliability of the software.
ii) Functionality describes what software does.	ii) Non-Functionality describes how software works.
iii) Concentrates on user requirement.	iii) Concentrates on user expectation.

iv) Functional testing takes place before Non-functional testing.	iv) Non-Functional testing performed after finishing Functional testing.

17) Software Testing Terminology:-

i) Regression Testing:- Testing conducts on modified build to make sure there will not be impact on existing functionality because of changes like adding / deleting / modifying features.

- User regression testing:- Testing only the changes/modifications done by the developer.
- Regional Regression Testing:-
  → Testing the modified module along with the impacted modules.
  → Impact Analysis meeting conducts to identify impacted modules with QA and Dev.
- Full Regression :- → Testing the main features and remaining part of the application.

Ex → Dev has done changes in many modules, instead of identifying impacted modules, we perform one round of full regression.

1) Re-Testing :- • Whenever the developers fixed a bug, tester will test the bug fix is called Re-testing.

• Tester close the bug if it worked otherwise re-open and send to developer.

• To ensure that the defects which were found and posted in the earlier build were fixed or not in the current built. Example :- → (1.0.1, 1.0.2)

→ Build 1.0 was released. Test team found some defects and posted.

→ Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

Example : Re-Testing Vs Regression Test

• An application under test has three modules namely Admin, Purchase, Finance.

Admin
↓
Purchase
↓
Finance

• Finance module depends on Purchase module.

• If a tester found a bug on Purchase module and posted. Once the bug is fixed, the tester

needs to do Retesting to verify whether the bug related to the Purchase is fixed or not and also tester needs to do Regression Testing to test the Finance module which depends on the Purchase module.

iii) iv) Smoke and Sanity Testing :-
Basic functional testing (both).
Both come into the picture after build release.

Smoke	Sanity
i) Smoke test is done to make sure the build we received from the development team is testable/stable or not.	i) Sanity test is done during the release phase to check for the main functionalities of the application without going deeper.
ii) Performed by both developer and tester.	ii) Performed by tester only
iii) build may be either stable or unstable.	iii) build is relatively stable.
iv) Done on initial builds.	iv) Done on stable builds.
v) Part of basic testing.	v) Part of regression testing.

v) Usually it is done everytime there is a new build release.

vi) It is planned when there is no enough time to do in depth testing.

vi) Exploratory Testing :- • We have to explore the application, understand completely and test it.

• Understand the application, identify all possible scenarios, document it then use it for testing.

• We do exploratory testing when the application ready but there is no requirement.

• Test engineers do the testing when there is no requirement.

Drawbacks :-

• Misunderstanding

• Time consuming

• No bug detect.

Any feature as bug / Any bug as feature.

vi) Adhoc Testing :- Testing

- Testing application randomly without any test cases or any business requirement document.
- Informal testing with an aim to break the system.
- Tester should have knowledge of application even though he doesn't have requirement/test cases.
- Unplanned activity.
  - i) No documentation
  - ii) No Test Design
  - iii) No Test Case

vii) Monkey / Gorilla Testing :-

- Testing application randomly without any testcases or any business requirement document.
- Informal testing with an aim to break the system.
- Tester do not have knowledge of application.
- Suitable for gaming application.

Adhoc	Monkey	Exploratory
i) No Documentation.	i) No Documentation	i) No Documentation.
ii) No plan	ii) No plan	ii) No plan
iii) Informal testing	iii) Informal testing	iii) Informal testing
iv) Tester should know application functionality.	iv) No need	iv) No need
v) Random testing	v) Random testing	v) Random testing.
vi) Intension is to break the application find out corner defects.	vi) Intension is to break the application/ - - - - - - - -	vi) Intension is to learn or explore functionality of application
vi) Any Applications	vi) Grooming Applications	vi) Any Application which is new to tester.

viii) Positive Testing :- Testing the application with valid inputs is called as Positive Testing.
It checks whether an application behaves as expected with positive inputs.

IX) Negative Testing :-

Testing the application with invalid inputs is called as negative testing. It checks whether an application behaves as expected with the negative inputs.

X) End to End Testing :- Testing the overall functionalities of the system including the data integration among all the modules is called end-to-end testing.

End to End testing
1) Login
2). Add new customer
3) Edit Customer
4) Delete customer
5) Logout.

XI) Globalization Testing :- Performed to ensure the system or software application can run in any cultural or local environment. (IISN)
Ex → Facebook. (Internationalization Testing.

06 SUNDAY

XII) Localization Testing :- Performed to check system or software application can run for a specific geographical and cultural environment

Ex → Baider.com (Chinese language) (Google).

18) Test Design Techniques :-

Used to prepare data for testing.
i) Reduce the data (test cases)
ii) More coverage.

Techniques :- Test data / case design Technique

i) Equivalence Class Partitioning (ECP) :-

Partition data into various classes and we can select data according to class then test. It reduce the number of test-cases and saves time for testing.

Ex:-

Enter a number : [ ] Allow digits from 1-500

Normal Test Data	Divide values into ECP	Test data using ECP
1	-100 - 0 → -50	(Invalid)
2	1 - 100 →	(Valid)
⋮	101 - 200 →	(Valid)
500	501 - 600 →	(Invalid)

i) Value check.
ii) Classify / devide / partition the data in multiple classes.

ii) Boundary Value Analysis (BVA Technique) :-

• BVA technique used to check boundaries of the input.

Ex → Enter a age. [ ] Allow digits from 18-35

Min = 18 Max = 35

Min - 1 = 17 (Fail) Max + 1 = 36 (Fail)

Min + 1 = 19 (Pass) Max - 1 = 34 (Pass)

Input domain testing :- The value will be verified in the text box/input fields. Here we use two techniques → i) ECP ii) BVA

iii) Decision Table :- / Cause Effect Table

• This technique will be used if we have more conditions and corresponding actions.

• In decision table technique, we deal with combinations of inputs.

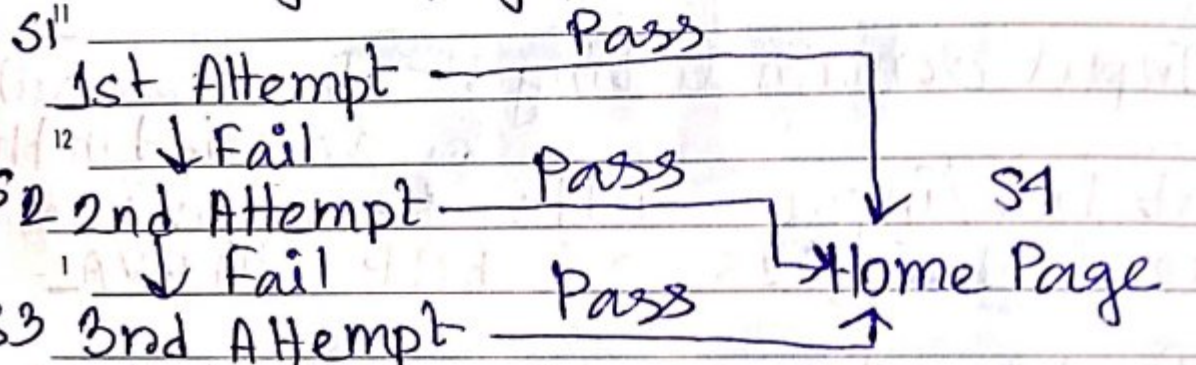
• To identify the test cases with decision table, we consider conditions and actions.

		TC1	TC2	TC3	TC4	TC5
Condition 1	Account already approved	True	T	T	T	F
Condition 2	OTP Matched	T	T	F	F	X
Condition 3	Sufficient money in Acc	T	F	T	F	X
Action 1	Transfer money	Execute				
Action 2	Insufficient amount	~~Execute~~	E		E	
Action 3	Block the transaction			E	E	X
		Valid TC	Negative TC			Invalid TC

iv) State Transition :-

- In state transition technique changes in input conditions change the state of the application.
- In tester can perform this action by entering various input conditions in a sequence.

Ex → login page of an application



State	login	Correct Password	Incorrect Password
S1	1st Attempt	S4	S2
S2	2nd Attempt	S4	S3
S3	3rd Attempt	S4	S5
S4	Home Page		
S5	Account locked		

v) Error Guessing :- Depends on tester's Analytical skills and

experiences. This technique is used to find bugs in a software application based on tester's experience.

Ex →

i) Submitting a form without entering values.

ii) Entering invalid values such as entering alphabets in the numeric fields.

SDLC → Software Development Life Cycle
Requirement Analysis, Design, Coding, testing, deployment, maintanance

19) STLC → Software Testing life Cycle.
i) Requirement Analysis
ii) Test planning
iii) Test Design
iv) Test Execution
v) Defect/Bug Reporting and Tracking
vi) Test closure.
cycle

1) Test Planning :- What to test
How to test
When to test
} Phase

Input :- Project Plan
Functional Requirements

Activities :- i) Identify the resources
ii) Team formation
iii) Test Estimation
iv) Preparation of test plan
v) Reviews on test plan
vi) Test Plan Sign Off.

Responsibility :- i) Test lead/Team lead (70%)
ii) Test manager (30%)
Outcome :- Test Plan Document

20) Test Plan Contents :- It is a document that describes the test scope, test strategy, objectives, schedule, deliverables and resources required to perform testing for a software product.

i) Use case :- It describes the requirement.
Use Case contains three items.
• Actor :- which is the user, which can be a single person or a group of people interacting with a process.
• Action :- which is to reach the final outcome.
20) Goal/Outcome : which is the

successful user outcome

ii) Test scenario :- A possible area to be tested (What to test)

iii) Test Case :- a) step by step actions to be performed to validate functionality of AUT (How to test)

b) Test case contains test steps, expected result and actual result.

Use Case	Test Case
i) Describes functional requirement.	i) Describes Test steps/Procedure
ii) Prepared by Business Analyst (BA)	ii) Prepared by Test Engineer.
iii) ~~What to test~~ Requirement, Define test scenario	iii) How to test

2) Test Suite :- It is a group of test cases which belongs to same category.

2) Test Case :- It is a set of actions executed to validate particular feature or functionality of your software application.

Test Case Contents :-

i) Testcase ID
ii) Testcase Title
iii) Description
iv) Pre-condition
v) Priority ($P_0, P_1, P_2, \ldots$) order
vi) Requirement ID
vii) Steps / Actions
viii) Expected Result
ix) Actual Result
x) Test data

23) RTM (Requirement Traceability Matrix) :- It describes the mapping of requirements with the test cases.

Purpose → To see that all test cases are covered so that no functionality should miss while doing software testing.

Parameters Include → i) Requirement ID
ii) Req Description
iii) Test Case ID.

Testers will check this.

24) Test Environment / Test Bed :- It is a platform specially build for test case execution on the

software product.
It simulates production/
real time environment.

25) Test Execution:- During this phase test team will carry out the testing based on the test plans and the test cases prepared.

26) Defects/Bugs:- Any mismatched functionality expected between expected and actual Result is called defects/bug.

Defect Reporting Tools:- I) Clear Quest.
II) Dev Track III) Jira
IV) Quality Center V) Bug Jilla

Severity → Describes the impact of the defect on the application
Priority:- It is related to defect fixing urgency (Importance of the defect)
Defect Category

Severity	Priority
I) Blocker (Can't proceed further)	I) P_1
II) Critical (functionality is not working)	II) P_2
III) Major (Undesirable behavior)	III) P_3
IV) Minor (Won't cause any major break down)	

Priority describes the importance of defect. Defect priority states the order in which a defect should be fixed.

P0 (High) :- The defect must be resolved immediately as it affects the system severely.

P1 (Medium) :- It can wait until a new versions/builds is created.

P2 (Low) :- Developer can fix it in later releases.

Priority

Severity	High	Low
High	Login is taking to the blank page	About us link is going to blank p
Low	After user is logged into application, he can see Home Page. But there is spelling mistake in Home page.	User opened contact page. Email ID has spelling mistake.

27) Defect Resolution :- After receiving the defect report from the testing team, development team conduct a review meeting to fix defects. Then they send a resolution type to the testing team for further communication.

28) Test cycle closure :- Stop the testing and application should not have bugs.
Deliverable → i) Test closure report
ii) Test metrics.

29) Test metrics :-

i) % of test cases executed :-
No. of test cases executed / Total no. of test cases written × 100 %

ii) % of test cases passed :-
No. of test cases passed / Total no. of test cases executed × 100

iii) Defect density :- No. of defects found / Size (no. of requirement)

• Defect removal efficiency (DRE) :-

{A/(A+B)} * 100

(Fixed Defects / (Fixed + Missed defects)) x100

7 Principle of Software testing.

i) Start software testing at early stages.

ii) Test the software in order to find the defects.

iii) Highly impossible to give the bug free software to the customer.

iv) Should not do exhaustive testing. Means we should not use same type of data for testing every time.

v) Testing is context based. Means decide what types of testing should be conducted based on type of application.

vi) We should follow the concept of Pesticide Paradox. Means, if you are executing same cases for longer run, they won't be find any defects.

vii) We should follow defect clustering. Means some of the modules contains

most of the defects. By experience, we can identify such risky modules. 80% of the problems are found in 20% of the modules.

Retesting vs Regression Testing :-

→ Executing the same test case by giving the number of inputs on same build called as Re-testing.

Regression testing is a type of software testing executed to check whether a code change has not unfavorably disturbed current features and functions of an Application.

QA Engineers need to spend lot of time and effort to perform retesting and regression testing. This can be reduced by using automation tools.

How automation tools work?
→ Manual test case → Manual execution
→ Manual to automation conversion →

automation script → execution automation test.

☑ What is Selenium :- Started on 2004
Selenium is web based automation tool.
No desktop or mobile

Application 3 types → i) desktop/ window based. ii) Web based, iii) Mobile

i) desktop/Window based application → No internet needed. install directly in laptop. MS Excel, Notepad, Powerpoint presentation.

ii) Web based application → internet is needed. installed remotely. net banking, browser is also needed

iii) Mobile application → whatsapp, linkedIn. In mobile on.

Selenium was originally developed by Jason Huggins in 2004, by Thoughworks Company. Selenium is open sources (means everyone is able to add some more features) 2022 Selenium is free of cost.

Selenium is collection of multiple components:
(IDE, RC, WebDriver, Grid).
* ↙ * *

Remote control (No use of it now)

Advantages :-
I) Open source tool and free.
II) Multiple operating systems. (Windows, mac).
III) Supports multiple browsers.
IV) Support multiple languages (Java, Python, C#, Ruby, Javascript etc).
V) Integrate third party tools in to Selenium.

Disadvantages
i) Cannot support window based applications (Autoit, Sikuli, Robot API).
II) Reporting not supported (TestNG, Extent reports)
III) Cannot support excel files (Apache poi)
IV) graphs, captcha

Selenium Versions :-
Selenium 1.x → IDE (Firefox) RC, Grid
" 2.x → IDE (Firefox), Webdriver, Grid
3.x →
4.x →

Manual Testing Project:

i) Project Introduction
ii) Understanding and Explore the functionality (FRS), Estimation
iii) Test Plan
iv) Writing Test Scenarios
v) Writing Test Cases and reviews
vi) Environment Setup and build deployment
vii) Test Execution
viii) Bug Reporting and Tracking
ix) Sanity testing, Re-testing, Regression testing
x) Test sign off.

i) Project Introduction :-

Product → E commerce product/ Application

E commerce → login, search for products/items, add them to cart, do payment, product will be delivered, returns the product etc.

Done by customer, frontend

Two parts → i) Front end (Public)
ii) Back end (Admin)

Internet → We can access it publicly

Intranet → Application which is used only by the organisers (backend)

Opencart

** Mockup Screen → Means a dummy screen which will design before actual software or actual application is designed. To imagine how the application looks like.

Agile model / Agile methodology / Agile process:

It is an iterative and incremental approach. Agile is an iterative and incremental process. Where requirements keeps on changing.
Iterative means same process repeating again and again.
Incremental means modules / features keep on adding on top of existing software.

Agile principles :-
i) Customer no need to wait for long time.
ii) We develop, test and release piece of software to the customer with few number of features.
iii) We can accept / accomodate requirement changes.
iv) There will be good communication between Customer, Business Analyst,

Developers and tasten.

Advantages :-

1) Requirement changes are allowed in any stage of development.
2) Releases will be very fast (Weekly)
3) Customer no need to wait for long time.
4) Good communication between team.
5) It is very easy model to adopt.

Disadvantages :-

Less focus on design and documentation since we deliver software very faster

Scrum :- Scrum is a framework through which we build software product by following Agile Principles.

Scrum team (5-9 members)

1) Product owner
2) Scrum Master
3) Dev Team. (Developer)
4) QA Team. (Tester)

1) Product Owner :- Define the features of the product. Prioritize features according to market value.

Adjust features and priority every

iteration, as needed.
Accept or reject work result.

2) Scrum Master :- The main role is facilitating and driving the agile process.

Scrum terminology :-

☑ User story :- A feature/module in software.

☑ Epic :- Collection of user stories.

☑ Product backlog :- Contains list of user stories. Prepared by product owner

☑ Sprint :- Period of time to complete user stories, decided by the product owner and team, usually 2-4 weeks of time. (Also known as Iteration).

☑ Sprint Planning meeting :- Meeting conducts with the team to define what can be delivered in the sprint and duration.

☑ Sprint backlog :- List of comitted stories by Dev/QA for specific sprint.

☑ Scrum meeting :-
Meeting conducted by Scrum Master everday 15 mins, called as scrum call/ scrum meeting. Q/ What did you do yesterday
Q/ What will you do today ?
Q/ Are there any impediments in your way ?

☑ Sprint retrospective meeting :-
conducts meeting after completion of sprint. The entire team, including both the Scrum Master and the product owner should participate

☑ Story Point :- Rough estimation of user stories, will be given by Dev and QA in the form of fibonacci series.

☑ Burndown Chart :- Shows how much work remaining in the sprint. Maintained by the scrum master daily. (Graph)

☑ Definition of Ready (DoR)
☑ Definition of Done (DoD)

JDBC

Database → Create, Insert, Update, Delete, Select

Java Database Connectivity is an API (application program interface) for the java programming language which defines how a client may access a database.

JDBC Stages :- 5 Stages
i) Load Driver Class ⟹ No need
ii) Build Connection
iii) Create statement
iv) Write Queries
v) Close Connection.