Testing:-

It is the process of Evaluating the system to check whether it is meeting the customer requirements (or) not

ER → Expected Result

AR → Actual Result

when ER = AR is means that the Application is defect free

when ER ≠ AR It indicates the presence of Defects in the Application

It is the process of Evaluating the system to check whether ER = AR

Gmail Example:-

Login page M_1	Inbox m_2	Draft m_3	Trash m_4
Google +	out box	Drive	hangouts
contacts	Starred	Duo	sent mail

Modules → High level Design [HLD]

Login page:-

User ID U_1	PWD U_2	Forgot PWD U_3.
sign up	sign in	create new account

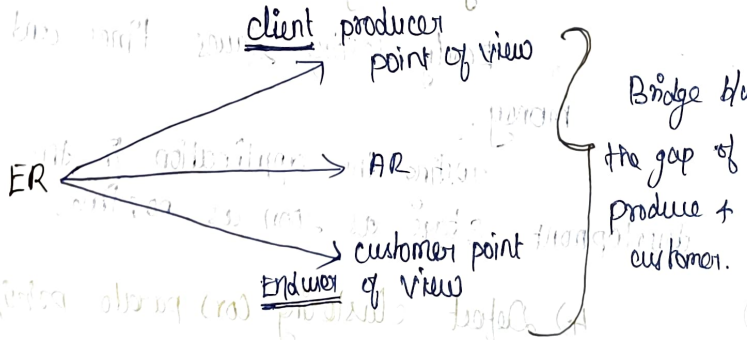
Low level Design [LLD]

System is the combination of modules & modules are the combination of units

(or)

units are combined to forms modules. modules are combined to form system units are nothing but codes.

Roles and Responsibility of Testing:-



QAT → Documentation, plan, & implement

QC → check ER = AR only

QAT → Quality Assurance Testing

QC → Quality Control.

General principle of testing:-

1) Testing source the presence of defects:-

Finding the defects in application does not means that the application is defective product

Testing helps to find the defect and Resolved.

2) Exhaustive testing is Impossible:-

Testing all the possible ways of combination. to the application is not at all possible.

3) Early Testing

Early Testing saves times and money.

Testing the application in the development stage as soon as possible.

4) Defect clustering (or) parato principle.

80% of issues are from 20% modules (or) codes.

The 20% of issue should be fixed Automatically the 80% of issues will be resolved.

5) pesticide pradocx:-

If any new functionality is the added to the Application the current test cases should be updated

6) Testing is content Dependent:-

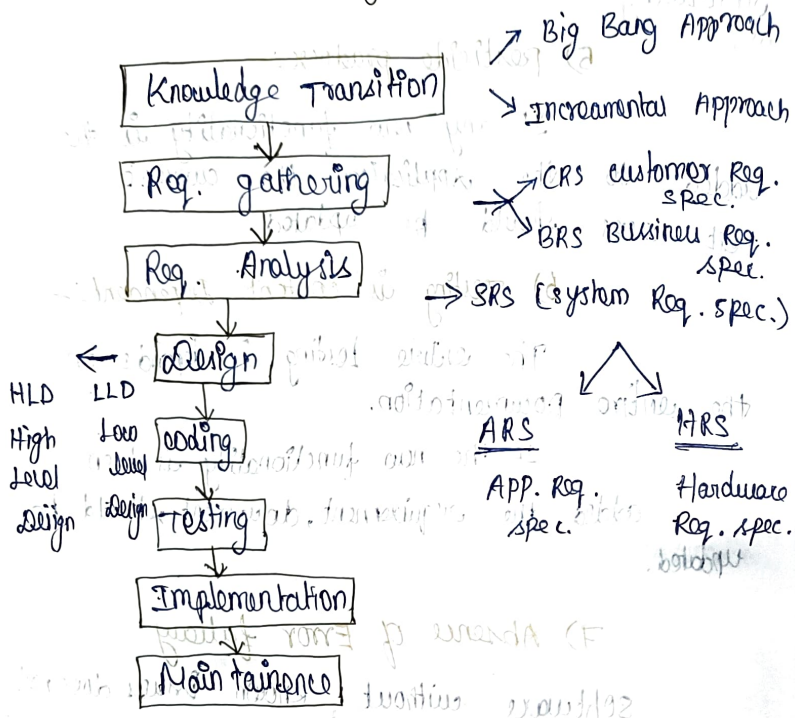
The entire testing is depends on the entire Doumentation.

If the new functionality is been added, the requirement, document should be updated.

7) Absence of Error fallacy

Software without known issues does not means the it is error free. finding and fixing some bugs doesn't granty the application is defect free product. Each and Every step of Testing should monitor carefully.

SDLC → Software Development life cycle



Knowledge transition

Ideas transfer from the customer location to the work location. It is carried out in two ways

(i) Big bang approach
(ii) Incremental approach.

(i) The ideas transfer for the entirely new product and new to the work location

(ii) Ideas transfer for the existing project for upgradation work.

Requirement gathering:-

It is the documentation work in which two types of documentation is carried out.

(i) CRS → customer Req. spec.
(ii) BRS → Business Req. spec.

(i) It is the documentation work submitted from the customer side to the work location and contains the customer requirement which will be in inorder form.

(ii) It is the Documentation done the help of CRS in this the customer requirement properly rearranged and transfer to the business motive.

Requirement Analysis:-

It is the documentation work. In which SRS is been documented and carried out in two ways.

(i) ARS → App. Req. spec.

(ii) HRS → Hardware Req. spec.

(i) It is the Documentation work which contains the platform require for the Application to be develop on the supporting tools required for the Application.

(ii) It is the documentation work which contains the details minimum hardware of required for the Application to run.

Design :-

It is the architectural documentation work carried out in two ways

(i) HLD

(ii) LLD

(i) It is the overall architectural of the entire system

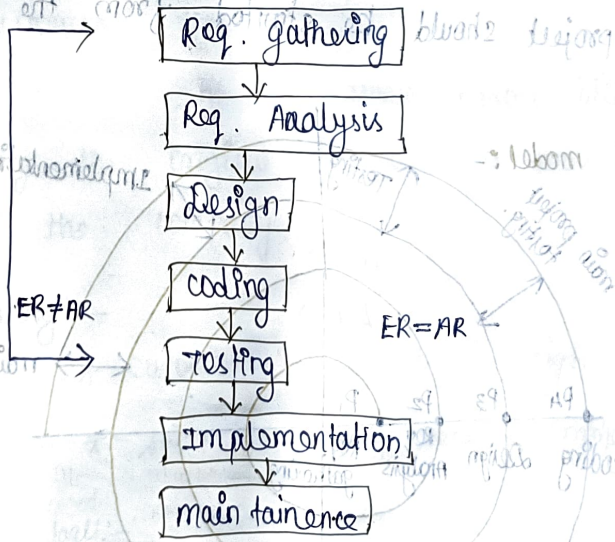
(ii) It is the brief architecture of the entire Design

Help of HLD & LLD Developer will start developing the code and the Tester (QA) start the test plan (or) test case

SDLC models

1) Water fall model
2) Spiral model
3) Agile model
4) Increamental model
5) Iterative model
6) VV model [verification & validation]

Water fall model :-



In this model after completing the finish phase then only the next phase will be started.

when ER ≠ AR then the entire project should be started from the first It is also called as traditional model.

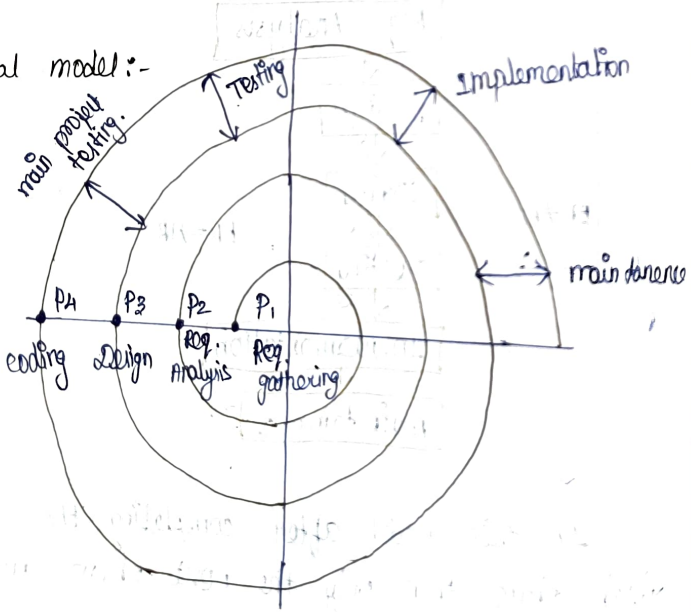
Advantages:-

It suits for static output projects and small projects

Disadvantages:-

* Customer involument is to low
* In testing ER ≠ AR the Entire project should be started from the first

spiral model:-



In this model each and every phase a prototype has been created and it is been tested

If the customer is been statisfied to the prototype then only the next phase will be started.

If the customer is not statisfied once again remarks will be done in the respective phase and the prototype will be created and given to the customer for testing.

In the implementation phase only the original product will move out of the company.

Advantages:-

* customer involument is high
* suits for highly secured projects

Disadvantages:-

time consuming and costly.

Agile model :-

Design coading
Testing M_1

Design. coding
Testing. Rework M_1, M_2

Design coding
Testing, Rework M_2 M_3.

Design coding Testing
Rework M_3, M_4

Rework M_1, M_2, M_3, M_4

* Let us consider a system consists of (4) modules M_1 M_2 M_3 & M_4

* M_1 will be develop & delivered to the customer for testing

* Simultaneously the development team will start working on M_2

* The customer will deliver the feedback of M_1

* The draw backs of M_1 will be development stage of M_2 & delivered to the customer as rework of M_1 & developed M_2

* This process is done till the end of the modules

Advantages:-

* Suits for masive as well as moderate projects

* Costomer Involvement too high

* Defect is been overcome in the development stage itself

Disadvantages:-

* Team leads should have clear idea about the entire project

Increamental model:-

M_1 → coding → Testing
M_2 → coding → Testing
M_3 → coding → Testing
M_4 → coding → Testing

Integration
↓
Implamentation
↓
main tainence

* In this for each and every module a separate developers and testers is been assigned.

* After developing & testing all the modules will be combined together and it will be Implemented.

Advantage:-

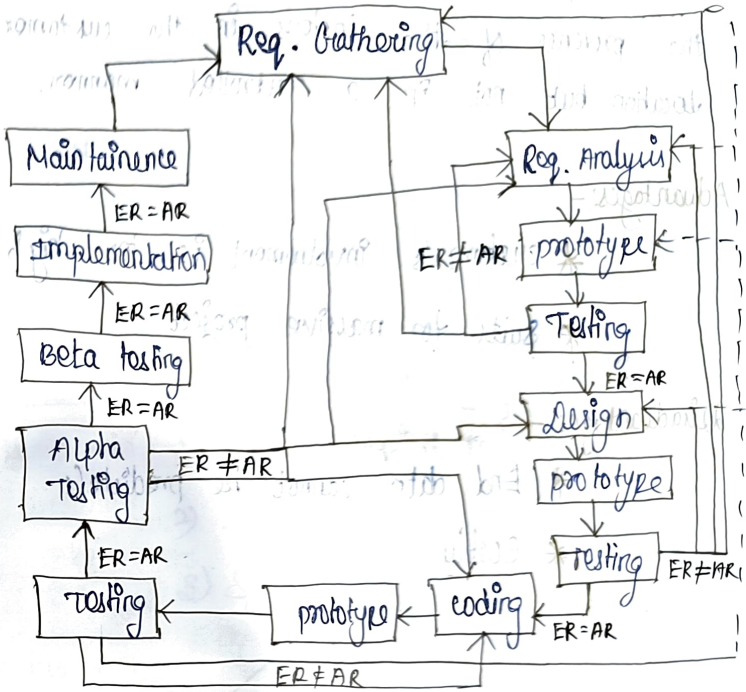
* It suits for the project which have less time duration

Disadvantages:-

* costly

* The projects Leads should have clear idea about the entire project

Iterative model:-



* It is also called as repetitive model

* Testing is repeated in each & every phase.

* In testing when ER ≠ AR we have access to verify the documentation in the previous phase.

Alpha testing

* Testers will test the app in the presence of developers in the developers location but in a controlled manner

Beta testing:-

* Customers will test the application in the presence of the testers in the customers location but not in a controlled manner.

Advantages:-

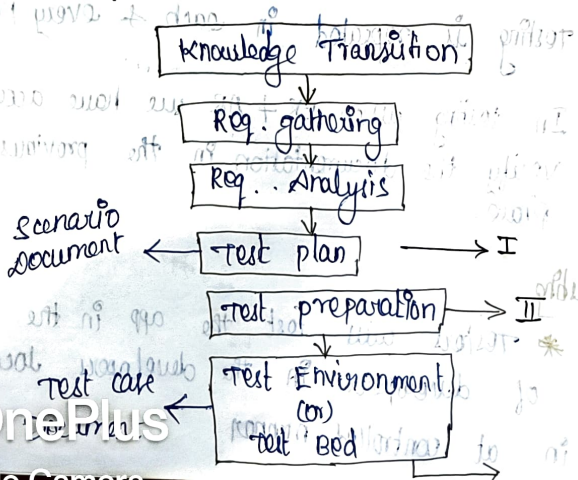
* customer's involvement is too high
* Suits for massive project

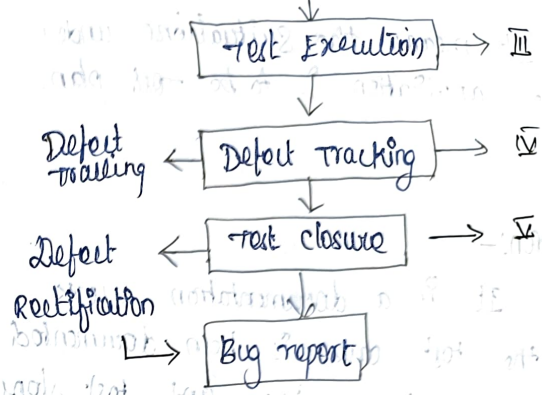
Disadvantages:-

* End date cannot be predicted.
* Costly

STLC

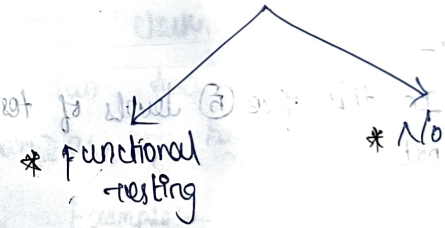
Software testing life cycle





Levels of Testing (or) Test Execution (⊗)

1) Unit Testing
2) Integration Testing
3) System Testing



4) System Integration Testing
5) (UAT) User Acceptance Testing

Test plan:-

It is the documentation work in which test scenario is been documented.

Scenario:- It means the situations under which the application is to be test plan.

Test preparation:-

It is a documentation work in which the test case is been documented which contains test procedure and test steps

Test Environment:-

It is the documentation work which contains the details of the test Environment in which the application is to be tested.

Test Execution:-

In this five (5) levels of testing is carried out

1) Unit Testing:-

In this codes will be tested and it is done by the developers.

2) Integration testing:-

It helps to check the connectivity between two unit (or) two modules

Example:- checking the connectivity User ID & password in gmail.

Note:- The user ID will navigate to the password if it is valid input. (or) if it is invalid it should restrict.

checking this process after connecting the username & password is known as Integration test

The next level of integration testing is Big Bang Integration testing

checking the connectivity between more than two unit (or) two modules known as big bang integration.

Example:- checking the connectivity between compose, sent mail, Draft in gmail.

If the mail is composed & sent if the mail is incomplete in Draft

In the Big Bang Integration is two types of approach carried out

(i) Top down approach.
(ii) Bottom up approach.

Top down approach:-

The testing starts from the initial stage to the final stage of the project if any one of the unit (or) modules is not been developed are dummy modules is used and is known as [STUB]

Bottom up:-

In this testing starts from the final stage to the initial stage of the project.

Dummy module used between is [Drive]

System testing:-

It is the process of Evaluating entire system to check whether (ER = AR)

It is carried out in two ways

(i) Functional testing
(ii) Non functional testing

Functional testing it is the process of Evaluating in behaviour of functionality of the application.

Non functional testing it is the process of Evaluating the system to check its capability

It is carried out in three ways.

(i) performance testing.
(ii) Load testing
(iii) stress testing

performance testing:-

Time taken to complete the particular functionality of the application.

Example:-

Let us consider a the web page is developed to respond one lakh users at the time.

Load testing:-

Testing the application within the range that is testing it with 10K, 20K, 30K users is known as load testing

stress testing:-

Testing the application above the range that is testing in which above 1 lakh is user is known as stress testing

[Additional testing:-

Reliability, Robustness, user friendly, maintainence.]

(API) (Application programming interface) (Application developer)

System Integration testing:-

checking the connectivity b/w two entirely difference system

Reason:-

If the developed application requires the other system help to do its functionality then the system integration testing will be performed

Ex:- checking the connectivity b/w ATM machine & Bank server

online shopping payment option.

user Acceptance testing UAT:-

It is the process of Evaluating to check weather the entire system it is meeting customer requirements (or) not. It is carried out in two ways.

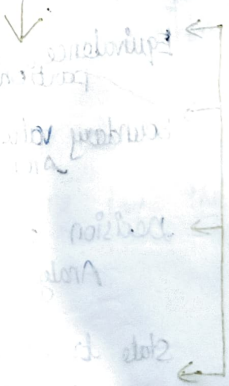
(i) Alpha testing
(ii) Beta testing

Defect Tracking:-

The identified defect in the test execution phase is been tracked and rectified and the report is been generated.

test closure:-

It is a collection of document from initial to the final stage of the product.



Types of Testing :-

Documents will be cross verified and checked

The application will be tested and executed.

Static (Documentation) (non-executable form)

Dynamic (Executable form)

Done by Developers:
- white box testing / structural testing / clear box testing

Done by testers:
- Black box / Behavioural testing / Opaque box testing

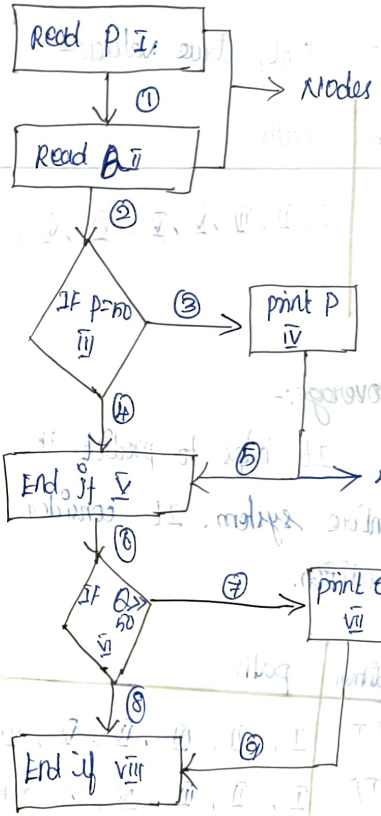
White box testing:
- statement coverage
- Decision coverage
- Basic path analysis
- Data table Analysis & codes

(Flow chart)

Black box testing:
- Equivalence partitioning
- Boundary value analysis
- Decision table Analysis
- State transition Analysis

* WB & BB is done in a single step it is known as Grey Box testing

White box testing :-



In this the codes will be tested and it is done by the developers.

statement coverage:-

It considers only True values to make sure all the branches in the entire system it is been covered

condition:- Only true values:-

condition	path.
TT	I, II, III, IV, V, VI, VII, VIII

Decision coverage:-

It helps to predict the possible testing path for entire system. It consider both true and false condition.

condition	path
TT	I, II, III, IV, V, VI, VII, VIII
TF	I, II, III, IV, V, VI, VIII
FT	I, II, III, V, VI, VII, VIII
FF	I, II, III, V, VI, VIII

Basic path Analysis:-

No. of Test path = No. of Edges − No. of nodes + 2(P)

P ⇒ unconnected nodes / hanging nodes

= 9 − 8 + 2(1)

= 3.

It helps to reduce the no. of testing path. identified in the decision coverage.

Data table Analysis:-

It helps to predict the state of the variable line by line inputs

It consists of 3 states.

Definition Occurance:-

The value has been defined to a variable

User Occurance:-

The value stored in the variable is used for a particular function.

predict Occurance:-

If the value stored in the variable is been compared than its known as (P&Q)

1. Read P;
2. Read Q;
3. if P = 50;
4. print P;
else
5. print Q;
else
6. Z = P+1;
7. X = Q+1
8. if Z > X
9. print Z
else
10. print X

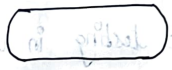
S.NO	DO	UO	PO
1.	P	–	–
2.	Q	–	–
3.	–	–	P
4.	–	P	–
5.	–	Q	–
6.	Z	P	–
7.	X	Q	–
8.	–	–	ZX
9.	–	Z	–
10.	–	X	–

use case diagrams:-

It is the pictorial representation of the entire system which explains the functionality flow of the entire system.

Basic symbols in use case diagram:-

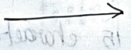

→ input / output


→ process

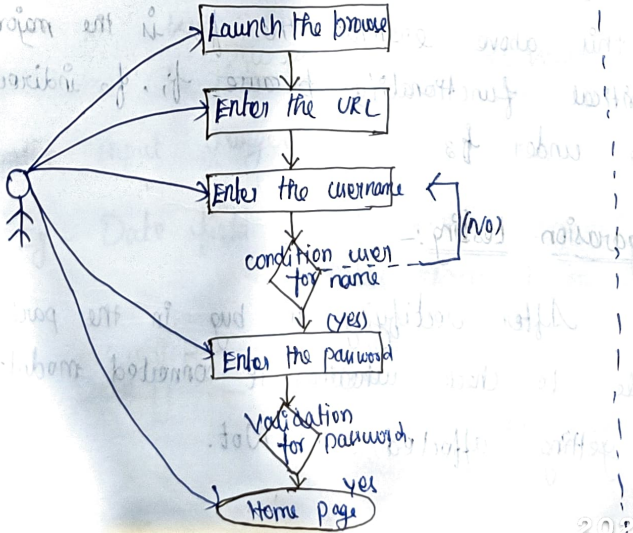

→ Decission.


→ Actor


→ system.


→ Data flow.

Gmail system:-



Special type of testings:-

1) Smoke Testing

It is the part of unit testing in which the major of critical functionality of particular application is been tested

Example:-

Let us conider username as pre-functionalities

→ f_1 it should accept alpha numeric & special characters

→ f_2 it should accept 10 to 15 characters length

→ f_3 it should accept only valid Input

In this above example, the f_3 is the major of critical functionality because f_1, f_2 indirectly false under f_3

2) Regression Testing:-

After rectifying the bug in the particular module to check whether it connected modules are getting affected (or) Not.

2) Sanity Testing:-

If the error is been identified in the Regression Testing the identified errors will be rectified and it will be whether the identified error in the Regression testing is been rectified (or) Not it is known as sanity Testing

Black Box Testing:-

It is done by the Testers in which the behaviour of the application is been tested

1) Equalance partioning:-

With the help of the requirement its helps to identify both the valid and Invalid input rangers.

Eg:- Date field :- It should accept Dates from 1 to 31

Date [1 – 31]

→ lower Boundary

→ Upper Boundary

Invalid	Valid	Invalid.
0	1	32
-1	2	33
-2	⋮	⋮
⋮	31	∞
∞		

Eg:- Age (18 – 88) Except (50 – 55)

Invalid	Valid	Invalid	Valid	Invalid.
17	18	50	56	89
16	19	⋮	⋮	⋮
⋮	⋮	55	88	∞
1	49			

Eg:- [10 – 15] characters - username field

Invalid	Valid	Invalid.
1	10	16
⋮	⋮	⋮
9	15	∞

2. Boundary Value Analysis:-

It helps to predict the valid and invalid test input's. The 3 terms in boundary Value Analysis are n, n+1, n-1;

* n – It occupies the value of lower boundary and upper boundary depending upon situations.

* (n+1) & (n-1) – occupies the place of valid & invalid scenarios depending on the situations

Eg:- Date field (1 – 31) → LB, → UB

Lower Boundary limit n = 1		upper Boundary limit n = 31	
Valid (n+1)	Invalid (n-1)	Valid (n-1)	Invalid (n+1)
2	0	30	32
Valid inputs [1, 2, 3 ..... 30, 31]		Invalid inputs :- [0 and 32 .....]	

Eg:- Age (18 – 88) except (50 – 55)

Lower Boundary Limits (n = 18)		upper Boundary limit n = 88	
Valid (n+1)	Invalid (n-1)	Valid (n-1)	Invalid (n+1)
19	17	87	89
Valid (n-1) n = 50	Invalid (n+1) n = 50	Valid (n+1) n = 55	Invalid (n-1) n = 55
49	51	56	54

Valid inputs:- (18, 19, 87, 49, 56.)

Invalid Inputs:- (17, 89, 51 & 54)

3) Decision table Analysis:-

It helps to predict the decision which is to be taken by the system depending on the Inputs's

Eg:- Gmail login Page:-

Inputs	Condition			
username	Valid	Invalid	Valid	Invalid
password	Valid	Valid	Invalid.	Invalid
Action.	Home page	Enter the valid UN	Enter the valid passwd	Enter the valid U.N.
V IV V IV	Blank V B IV B			
V IV IV V	Blank B V B IV			

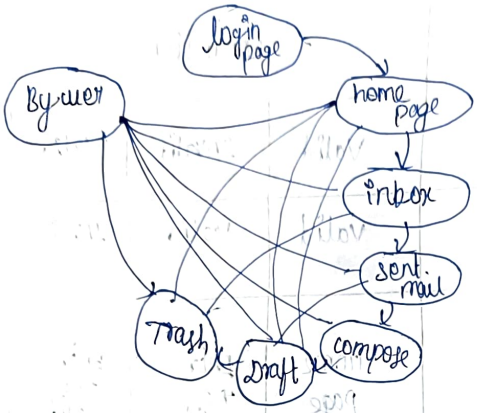
No. of combinations = (no. of condition)^(no. of inputs)

= $(2)^2 = 4$

= $(3)^2 = 9$.

State Transition Analysis:-

It helps to predict the state shiftins of s/m dependings on the input's



present state	Inputs	Next state
Login page	valid Inputs	Home page
home page	click on inbox	Inbox
home page	click on compose	compose.
home page	click on Draft	Draft
home page	click on Trash	Trash.
home page	click on logout	logout
sent mail	click on compose	compose.
sent mail	click on Draft	Draft
sent mail	click on logout	logout

Realiability → நம்பக தன்மை.

Robustness → secure safe in hackers.

user friendly → to clear follow the application in the user side.

Test case Description content

Test case ID, Test case description, pre-requisite, Test step, Test data, Expected Result, Actual result, Result.