MQ

SURESH REDDY NOTES

KONARK XEROX Cell:9949090558
Nagasuri plaza Ammeerpet Hyd

cher's
gn /
narks

** Why did u choose Testing ?

1. scope of getting job is very very high.
2. No need to depend upon any technology.
3. Testing will be there forever.
4. I want to be consistent throughout my life.

* Why Explicitly the test engineers are been recruited by the softwaare companies ?

Now a days only ?

1. one person cannot perform two tasks efficiently ✓ ×
   at a time.
2. sentimental attachment.

* who can perform testing ?

Any graduale who has creativity can perform testing.

19/12

* What exactly we require to get a job ?

1. Stuff
2. confidence
3. communication skills — R+W+S+L
4. Dynamism.

Project

Project is something that is developed based on particular costumer's requirements and for his usage only.

Product :-

Product is something that is developed based on the company's specifications & can be used by multiple customers.

Note :- usually, the company will decide the requirements based on the generic requirements of so many customers in case of products.

QUALITY

Classical def. of quality :

Quality is defined as justification of all the requirements of a customer in a product.

Defect :

Defect is defined as deviation from the requirements.

Latest def. for quality :-

Quality is defined as not only the justification of the requirements but also the presence of value (user-friendliness).

Note :- Quality is not defined in the product, it is defined in the customer's mind.

TESTING

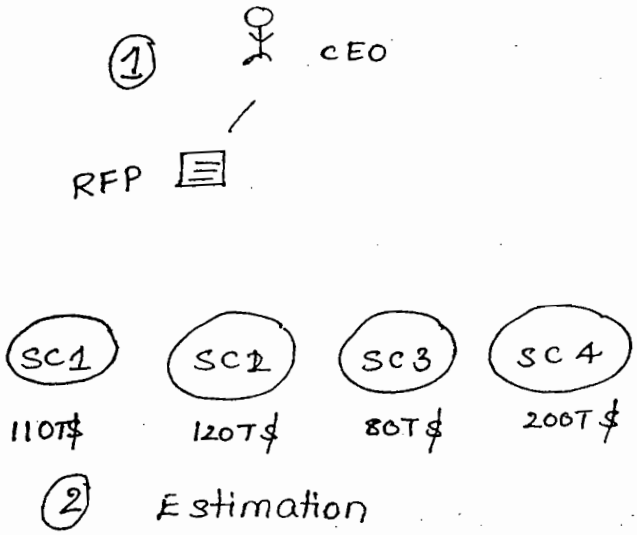
Testing is a process in which the defects are identified, isolated (seperated), subjected for rectification and ensured that the product is defect-free inorder to produce the quality product in the end and hence customer satisfaction.

defect profile
↳ (Isolated)

28/12

Bidding The project :-

Bidding the project is defined as Request For proposal, Estimation and signing off.



Kick-off meeting :-

It is an initial meeting conducted in The software company soon after The project is signed off inorder to discuss The overview of The project and also to select a project manager for that project.

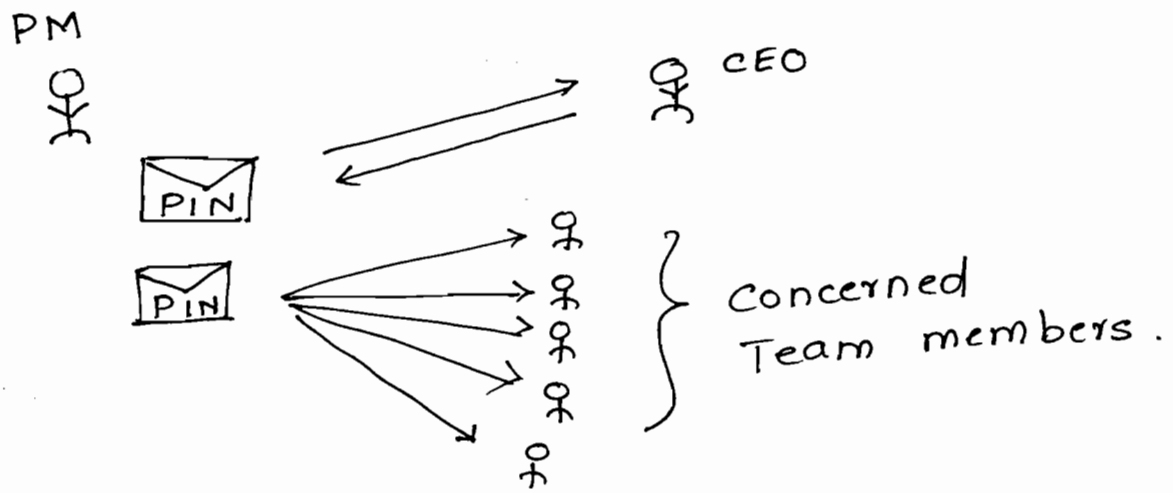
Usually, High level management, project Manager, Technical Managers, quality/ Test manager, Dev. leads & Test leads will involve in this meeting.

Note :- usually any kind of start up meetings can be called as kick-off meetings.

PIN (Project Initiation Note) :-

PIN is a mail prepared by the project manager and sent to the CEO of the software company inorder to get the permission to start the project development.

Once the permission is given, he will prepare another mail & will send it to all the concerned team members inorder to intimate them that they are about to start the project. Even this mail is also known as PIN only.



24/12

SDLC (Software Development Life cycle) :-

Software dev. life cycle contain 6 phases.

1. Initial (or) requirements phase
2. Analysis phase
3. Design phase
4. Coding phase (or) programing phase.
5. Testing phase
6. Delivery & maintanence phase.

Initial phase :-

Task : Interacting with the customer and gathering the requirements.

Roles : Business Analyst (BA)
Engagement Manager (EM)

Process : First of all the BA will take an appointment from the customer, collects the Template from the company, meets the customer on the appointment date & gathers the requirements, with the help of the template. Finally comes back to the company with the requirements document.

BA

The EM will go through the requirements & if at all he finds any extra requirements, then he deals with the excess cost of the project. If at all he finds any confused req's then he will ~~all~~ ask the concerned team to develop a prototype, go to the customer with that prototype, demonstrates the prototype, gathers the clear req's & will finally hand over the requirements document to the BA.

EM

↓

FRS - fnl req. spe. BDD — Business Design Doc
BRS - Bus. " " BD — " Doc.
CRS — client.

Note: some companies may maintain 2 docs in the initial phase. one is for overall Business flow Information & the other is for detailed fol info.

proof : The proof doc. of Initial phase is requirement document.

This document will be called with foll. names by different companies.

or SRS ↓ software

1) FRS (Functional Requirement specification)
2) BRS (Business Requirement specification)
3) CRS (customer / client Requirement specification)
4) URS (User Requirement specification)
5) BDD (Business Design Doc)
6) BD (Business Doc)

Note contd :

Some companies will maintain both the information in a single document.

***

Template :

It is defined as a pre-defined format which is used for preparing any document very easily and perfectly.

***

prototype :

prototype is roughly and rapidly developed model which is used for demonstrating to the client inorder to gather the clear requirements and also to win the confidence of a customer.

Analysis phase :-

Tasks :- 1. Feasability study
2. Tentative planning
3. Technology selection & Environment confirmation.
4. Requirement Analysis

Roles :- System Analyst (SA)

Support SA. { Project Manager (PM)
Technical Team Manager (TM)

Process :-

1. Feasability study ;

Possibility

Feasability study is a detailed study of the requirements inorder to check whether the requirements are possible within the avaidable time, budget and resources.

2. Tentative planning :

Temporary

The resources as well as the time will be temporarily planned here in this section.

3. Technology selection & Environment confirmation ;

The list of all the technologies that are reqd. to accomplish this project successfully and the details of the environment which will be suitable for this application will be

4. Requirement Analysis :-

The list of all the requirements like human resources, softwares & hardware reqd. to accomplish the project successfully will be clearly analysed and listed out here in this section.

Proof :-

The proof document of the Analysis phase is system Requirement specification (SRS)

26/12

Design phase :

Tasks :- 1. High level designing
2. Low level designing

Roles :- Cheif Architect (CA) will do the high level designing

Technical Lead (TL) will do the Low level designing.

process :- The cheif Architect will divide the whole project into modules by drawing some diagrams (Data flow diagrams) using unified modelling Language (UML). The technical lead will divide the modules into sub modules by drawing some diagrams using same UML.

The Technical Lead & his team with the support of cheif Architect will design the gui part of the application & also develops the

Proof :- The proof document of the design phase is Technical Design Document. (TDD)

Pseudo code :-

Pseudo code is a set of english statements which will make the developers more comfortable while developing the actual code.

Coding phase :-

Task :- Programming / coding.

Roles :- Programmers / developers.

Process :- The developers will develop the Actual source code (ASC) by following the coding standards (like proper Indentation, proper color codings, proper commenting and etc.) and with the support of Technical Design Document.

proof :- The proof document of the coding phase is source code document (SCD).

Testing phase :-

Task :- Testing.

Roles; Test Engineers.

Review report

Process :-

1. The test engineers will be receiving the req's Doc. and start understanding the req's.
2. While understanding the req's., if at all they get any doubts, then they will list out all the doubts in the requirements clarification Note (RCN).
3. The RCN will be sent to the author of the req's. document for clarifications
4. Once the clarifications are given, and after understanding all the req's. clearly, then they will take the testcase template and write the testcases.
5. Once the first build is released, they will execute the testcases.
6. If at all any defects are found, they will list out in the defect profile.
7. The defect profile will be sent to the development department and they will wait for the next build.
8. once the next build is released, they will reexecute the reqd. testcases.
9. If any defects are identified, then they will update the defect profile, send it to the dev. department and wait for the

next build.

10. Once the next build is released, they will repeat the process, till they feel the product is defect free.

proof :- The proof of Testing phase is quality product.

***

Test case : Test case is an idea of a Test engineer to test something of an application based on the requirements.

***

GUI : It is an interface ~~developed~~ between the user and something developed with graphics inorder to make the user comfortable to interact with that something.

27/12

Delivery & Maintanence phase :-

Delivery :-

Tasks :- Delivering the software.

Roles :- Deployment Engineers (or) Installation Engineers

process :- The Deployment Engineers will go to the client's place, and install the application into the customer's environ-

proof :- After delivering the application, The official agreement will be signed off. That document will be The proof for delivery phase.

Maintenance :-

Once The application is delivered, The customers will start using it, while using if at all they face any problems, Then That problems will become task, based on that task corresponding roles will be appointed, they will solve The problem by defining The process. This type of maintenance is known as normal maintenance.

But some clients may be requesting for continuous maintenance also. In That situation a team of members from our company will be working in The client's place regularly inorder to take care of The software. This type of maintenance is called continuous maintenance.

*** Where exactly Testing comes into picture?
which sort of Testing u r expecting ?
How many sorts of Testings are There ?

There are 2 sorts of Testing.

a) unconventional Testing
b) conventional Testing

unconventional Testing :-

It is a sort of testing in which The Quality Assurance people will check each and

every outcome doc. & company's process right from the initial phase of SDLC upto the end of SDLC.

Conventional Testing:

It is a sort of Testing in which the test engineers will be testing the application or its related parts in the coding and Testing phases of SDLC.

Testing Methodology :-

Testing methods (Testing Techniques) :-

There are 3 methods of Testing.

1. Black box testing
2. white box testing
3. grey box testing

Black box testing :-

If one performs testing only on the fnl part of an application without having the knowledge of structural part then that method of testing is known as black box testing.

usually test engineers will be involved in black box testing.

White box testing / glass box testing / clear box testing:

If one performs testing on the structural part of an application then that method of testing is known as white box testing.

usually, developers or white box testers will be involved in white box testing.

Grey box testing :-

It is a method of testing in which one will perform testing on both the fnl part as well as the structural part of an application.

usually the test engineers who has the additional knowledge of structural part will be involved in grey box testing.

29/12

Levels of Testing :-

There are 5 levels of Testing.

1. unit level Testing
2. Module level Testing
3. Integration level Testing
4. system level Testing
5. User-acceptance level Testing

unit level Testing :-

↓

Def:- unit is defined as a smallest part of an application. usually the programs are called ~~the~~ as units.

In this level of Testing, each and every program developed will be tested as well as some programs will be combined and will be tested by the white box testers.

As the white box testers are testing the programs, this level of testing falls under white box testing.

Module level Testing :-

Def: Module is defined as a group of related features to perform a major task.

In this level, The module will be released to The testing Dept. and the test engineers will be testing the fnl part of that module.

As the fnl part is been tested, The module level testing will fall under black box testing.

Integration level Testing :-

Interface is a program which is used for linking the modules.

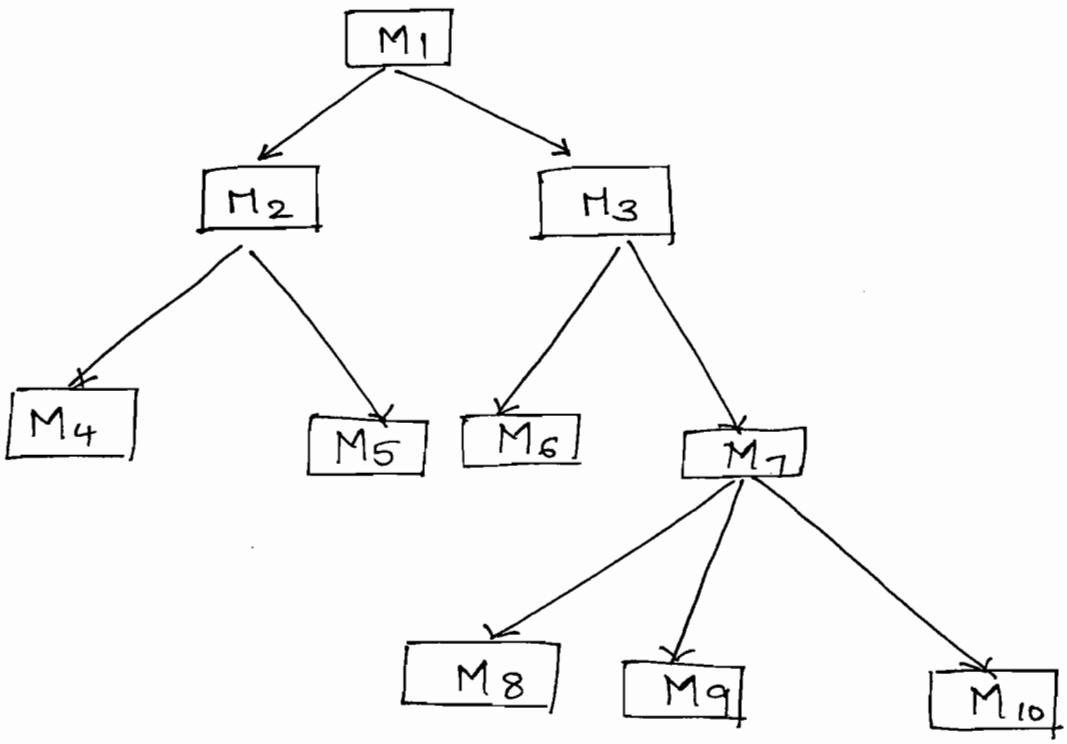
In this level The developers will be developing the interfaces inorder to integrate the modules, those interfaces will be tested inorder to check whether they

As some programs are been tested, integration level testing will fall under white box testing.

Types of approaches been followed by the developers while integrating the modules:

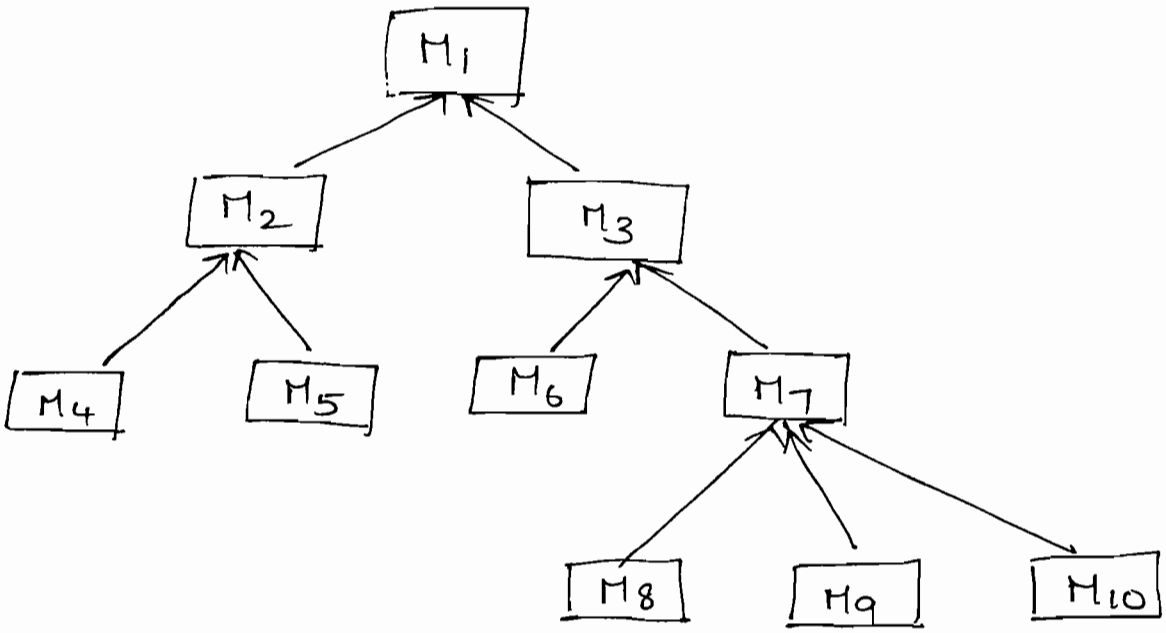
a) Top down approach:

In this approach, the parent modules will be developed first and will be integrated to the corresponding child modules.



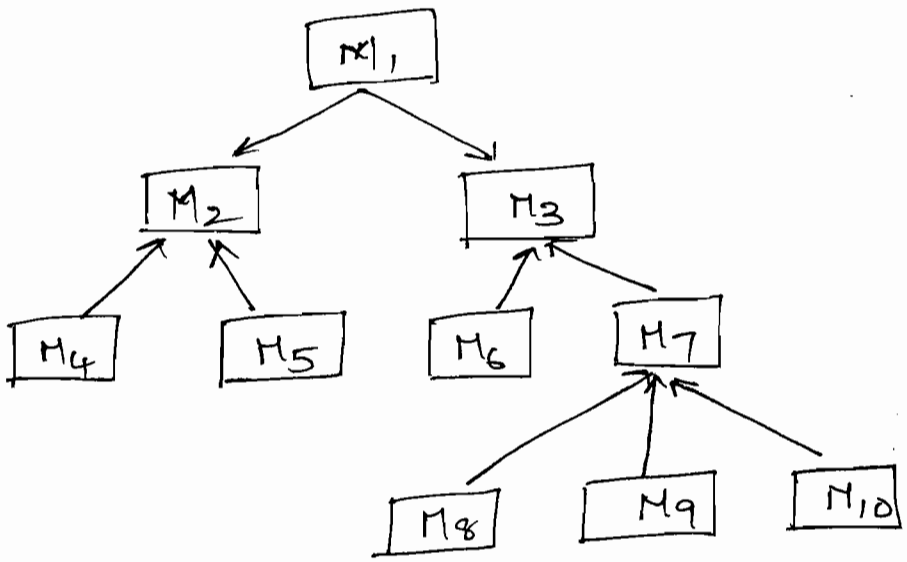
b) Bottom up approach:

In this approach, child modules will be developed first and then integrated back to the corresponding parent modules.



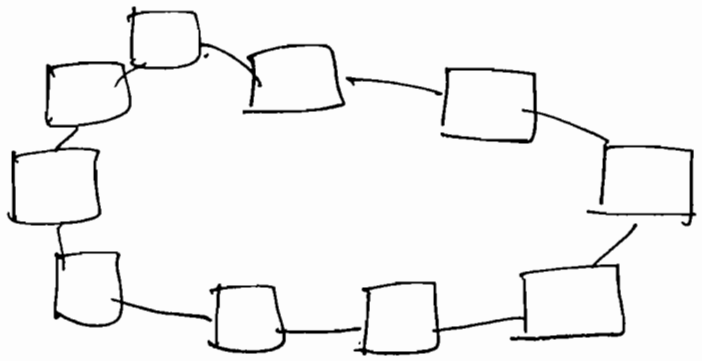
c) Hybrid approach (or) sandwich approach:

It is a mixed approach of both the top down and bottom up approaches.



d) Big bang approach :-

In this approach, one will wait till all the modules are ready & will integrate them at a time finally.



STUB AND DRIVER :-

While integrated ting The modules in Top down approach, if at all any mandatory module is missing then That module is replaced with a temporary program known as STUB.

While integrating the modules in bottom up approach, if at all any mandatary module is missing Then that module is replaced with a temporary program known as DRIVER.

System level Testing :-

In this level, the test engineers will perform different types of testing on The system. one among Those very imp. type of testing is 'system Integration Testing'.

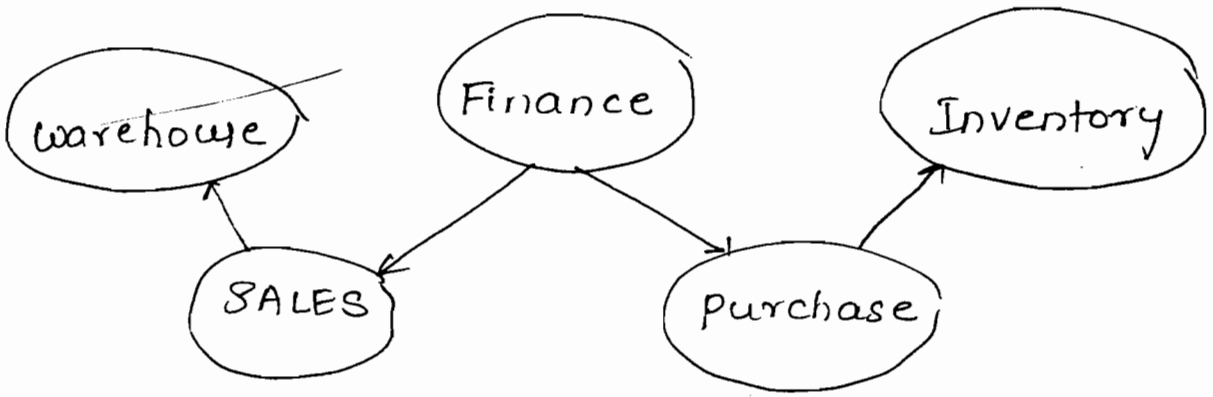
Apart from this type of testing, some other types like Load Testing, performance testing, stress testing, usability testing etc. will be conducted in this stage.

As the black box testers are conducting the testing, The system level testing will fall under black box testing.

System Integration Testing :-

It is a type of testing in which one will perform some actions in a module & check for the reflections in all the related modules.

Ex: ERP project.



User acceptance level Testing :-

In this level, the test engineers will check the user desired areas in the presence of user inorder to make him accept the application.

As the test Engineers are performing the user acceptance testing, it will fall under black box testing.

Environments

Environment is a combination of presentation layer, Business layer and database layer which can hold The presentation logic, Business logic and database logic. (which can hold something.)

(or)

Environment is defined as a group of hardware components with some softwares where we can install The presentation logic, Business logic & database logic.

Types of Environments :

There are 4 types of environments

1. stand alone environment-
(or)
1-tier architecture

2. client server environment
(or)
2-tier architecture

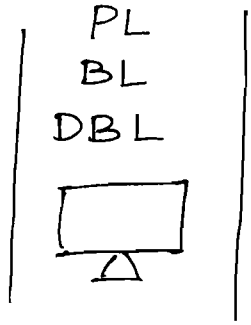
3. web environment
(or)
3-tier architecture

4. Distributed environment
(or)
n-tier architecture.

Stand-alone environment :-

This environment contains only one tier. All the three layers i.e., presentation layer, Business layer and database layer will be present in that tier only.

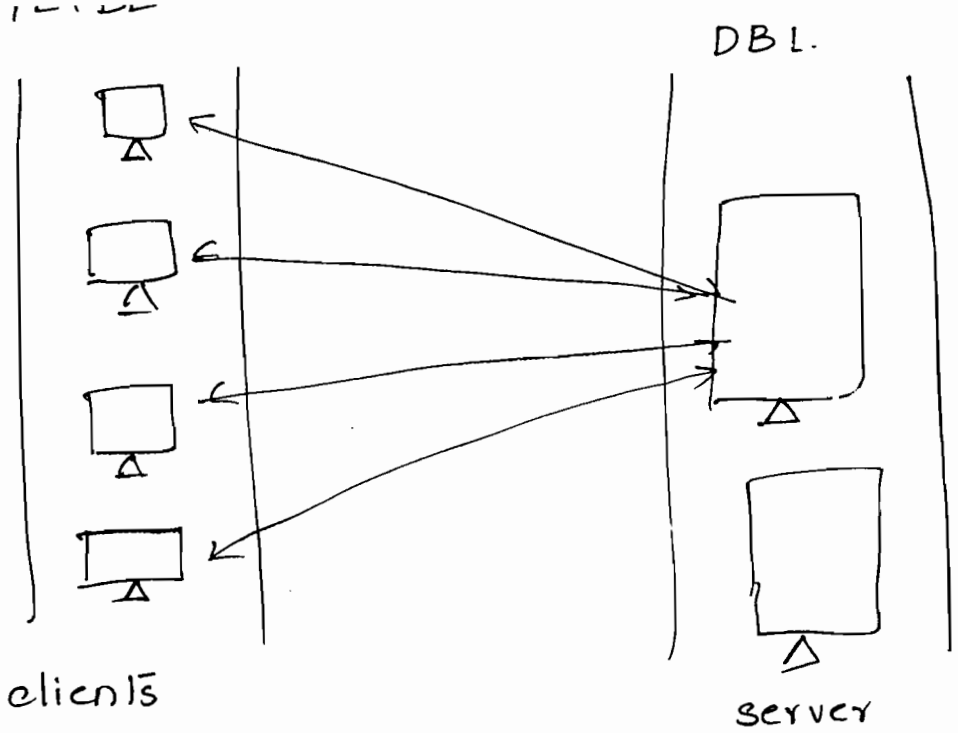
*** If at all the application need to be used by a single user in a single machine or a single place then the best environment suggested is stand-alone environment.



Client-server environment :-

This environment contains 2 tiers. one is for clients and other one is for servers. (no of servers) or 1 # The presentation layer and the business layer will be present in the clients and the database layer will be present in the servers.

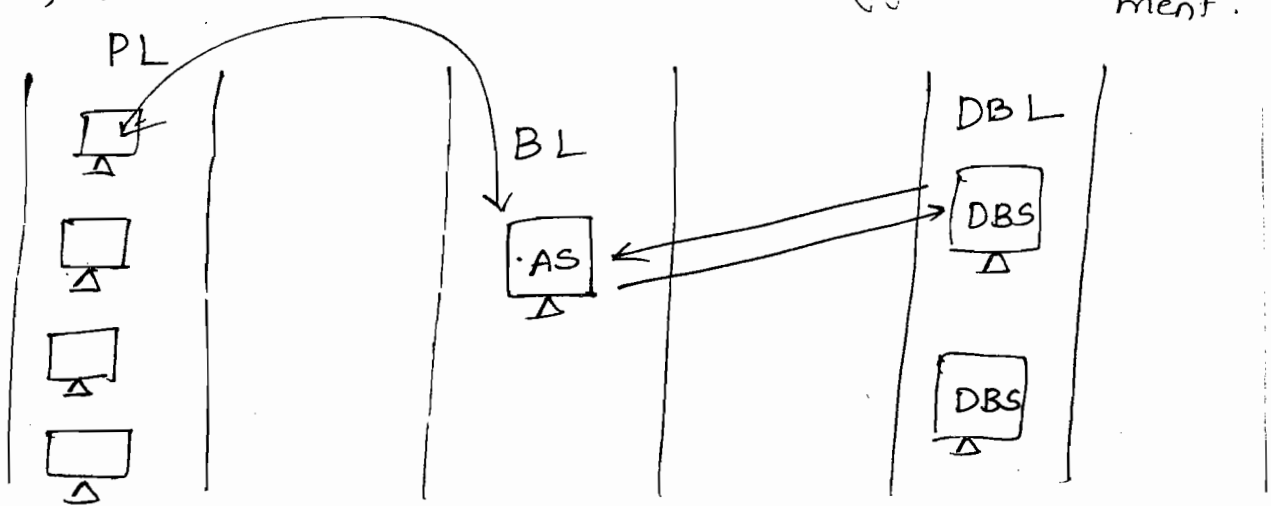
*** If at all the application need to be used in a single building or a single place or in a single city by multiple users, application need to be accessed very fastly and if there is no problem with security then the best environment



Web environment :

This environment contains 3 tiers. one is for clients, the middle one is for application server and the other ~~tiers~~ tier is for database servers. presentation logic will be present in clients, Business logic will be present in application server and database logic will be present in database serves.

If at all the application need to be used all over the world by limited no of users, then this is the best suggested environment.

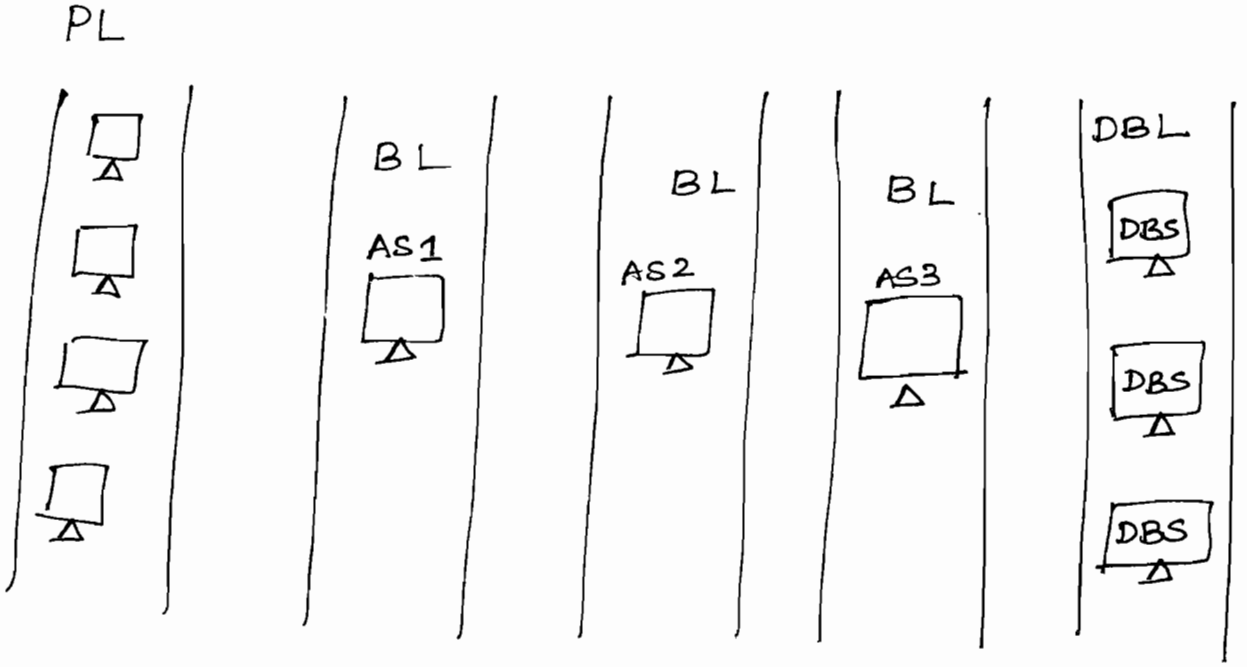


* server

Distributed environment

This environment is similar to web environment but no. of application servers are introduced in seperate tiers. To distribute The Business logic so that The load will be distributed and performance will be increased.

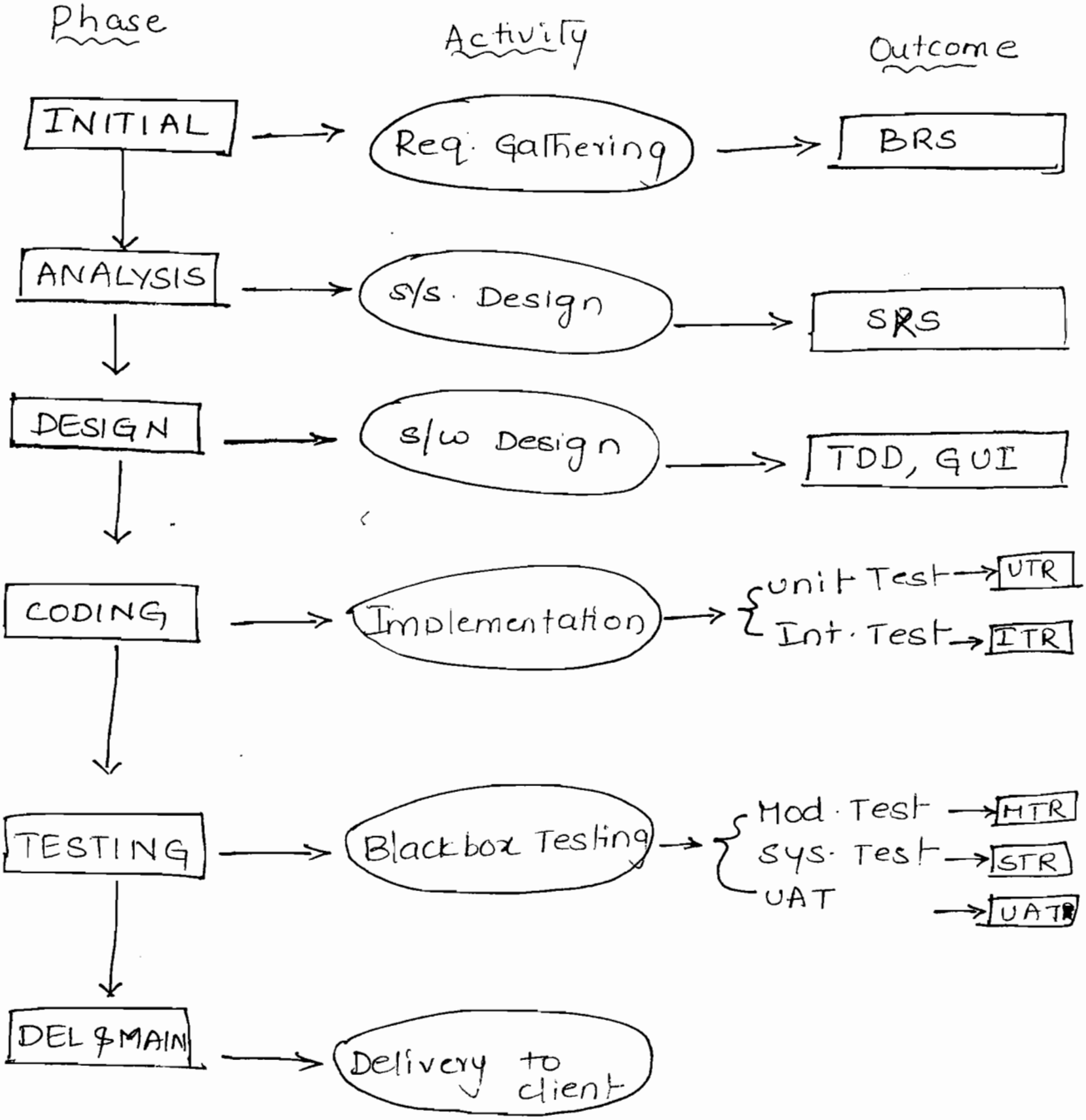
** If at all the application need to be used by more no. of people, all over The world then this is the best suggested environment.



5 tier Architecture

thin & thick client :-

Software process development model :-



WATERFALL MODEL

→ who will decide which model to follow ?

Advantages :-

1. It is a simple model.
2. project monitoring & maintenance is easy

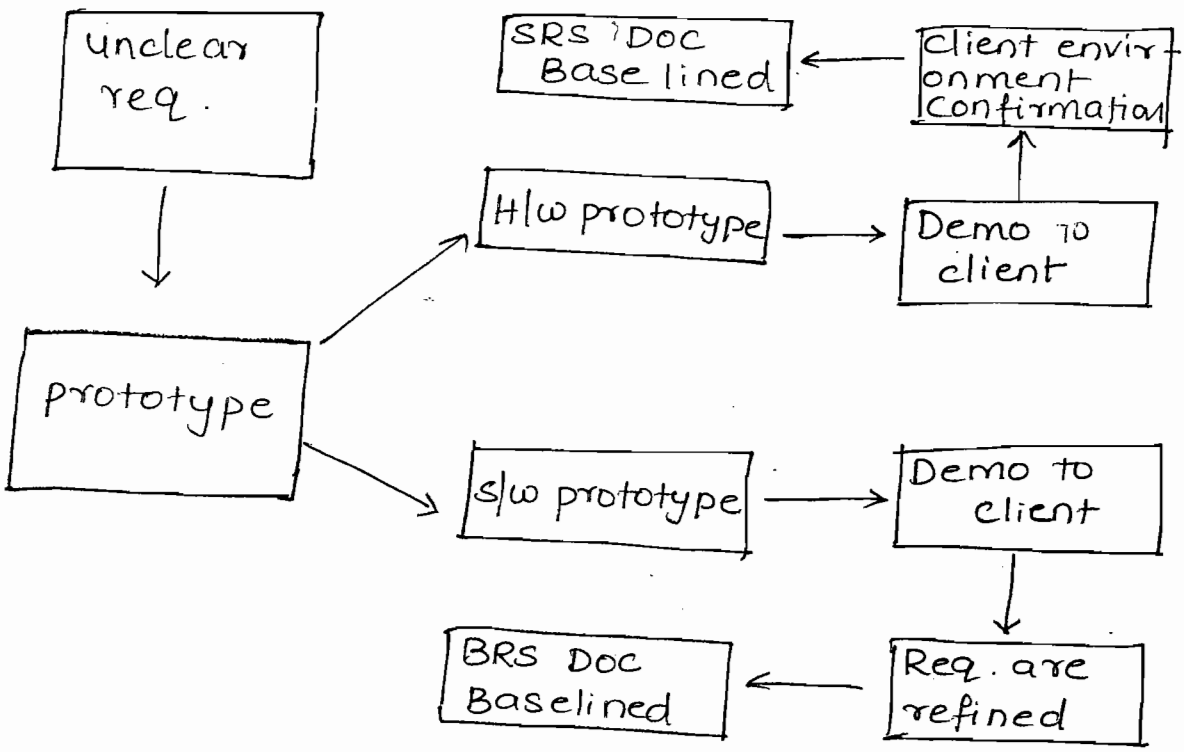
Drawbacks

1. cannot accept the new requirement in the middle of the project.

** Note :- This model is suitable for clear customer

31/12

PROTOTYPE MODEL



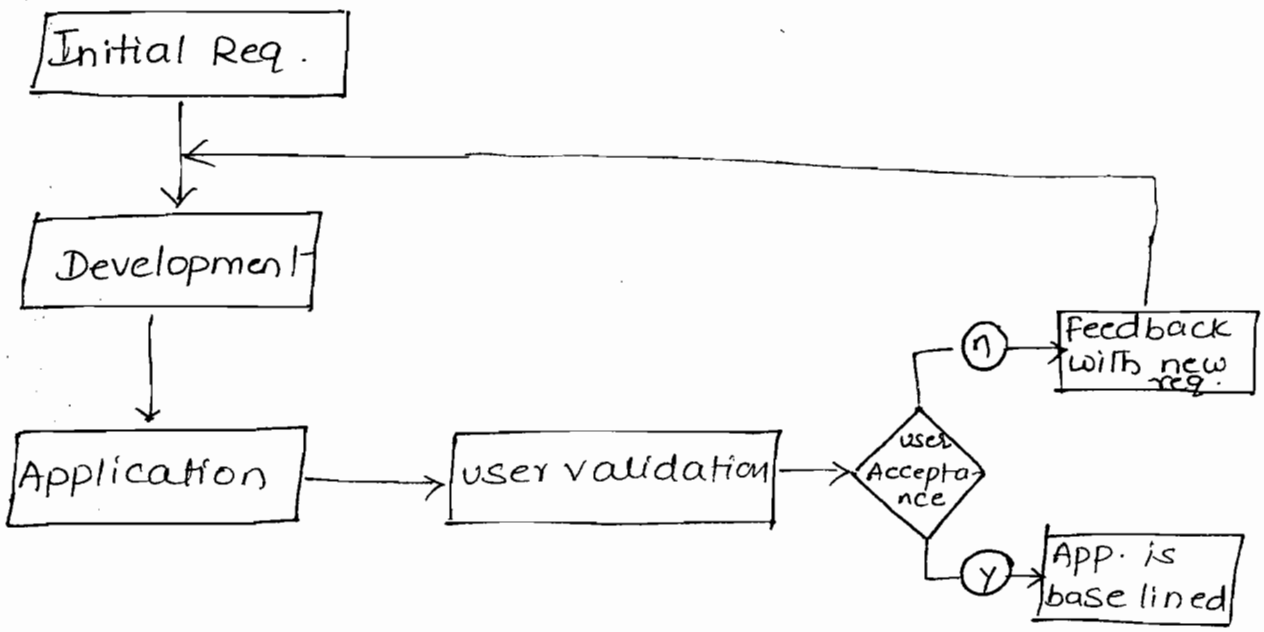
Advantages :-

1. Whenever The req's are not clear, this is The best suited model.

Drawbacks :-

1. It is not a complete model.
2. slightly time consuming model.
3. prototypes need to be built on company's cost.
4. customer may limit his req's by sticking to the prototype.

EVOLUTIONARY MODEL.



Advantages :-

giving req's in instalments.

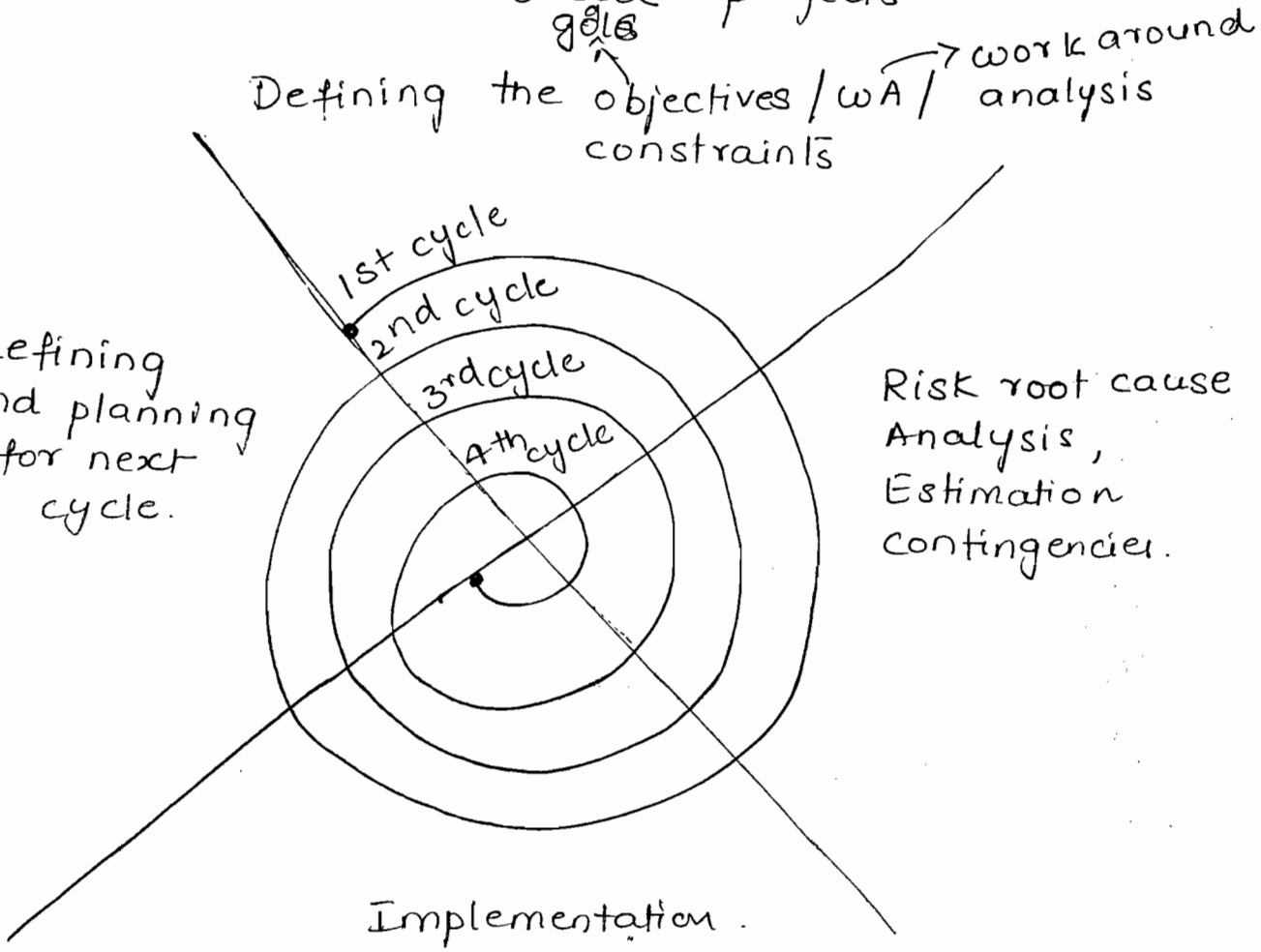
1. whenever The customer is evolving The req's., Then this is The best suited model.

Drawbacks :-

1. Deadlines are not clearly defined.
2. Time consuming model.
3. costly model.
4. No Transperancy.
5. project monitoring & maintenance is difficult.

SPIRAL MODEL

→ risk-based projects



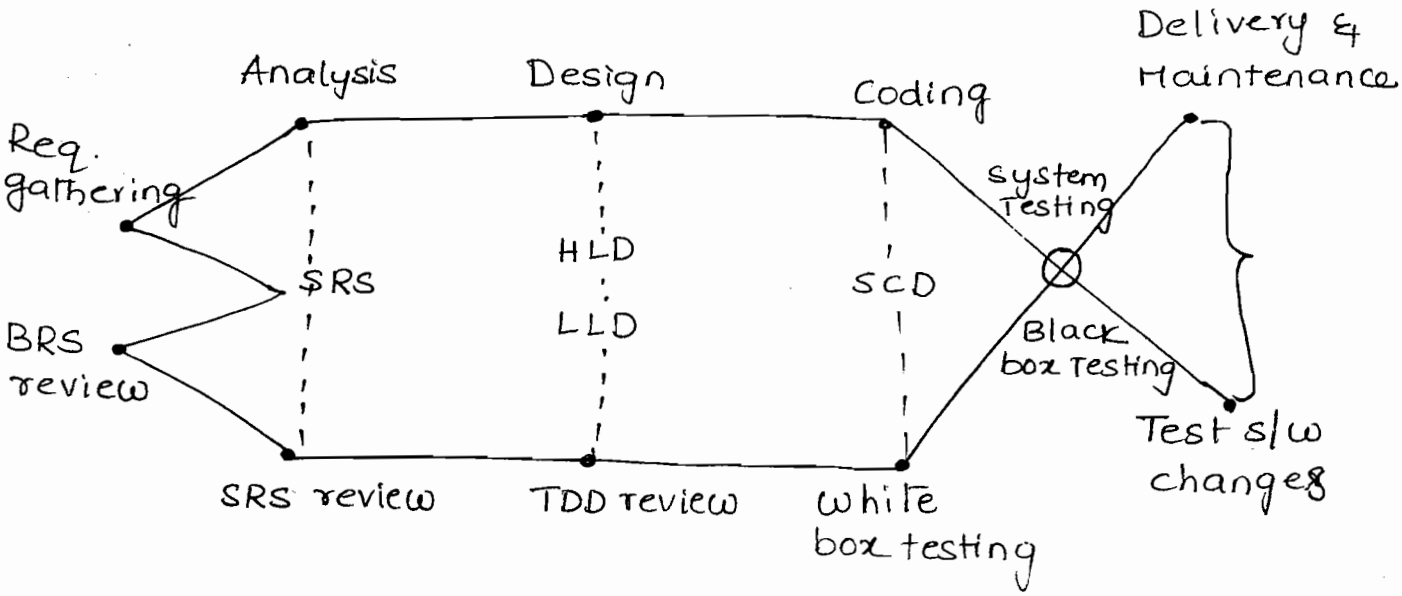
Advantages :-

1. whenever the project is highly risk based, then this is the best suited model.

Drawbacks :-

1. Time consuming model.
2. project monitoring and maintance is a difficult task.
3. Risk root cause analysis is not a easy task
4. It is a costly one.

FISH MODEL



Advantages :-

1. As both verification and validation are done, The outcome will be a quality product.

Drawbacks :-

1. Time consuming model.
2. costly model.

Verification :-

Verification is a process in which one will check each and every role in the organisation inorder to confirm whether they are working according to the company's guidelines or not.

Validation :-

Validation is a process of checking the developed application or its related parts inorder to confirm whether they are working according to the req's or not.

Note :- Verification is for checking the process and validation is for checking the product.

2/1/09

V MODEL

stands for
Verification
&
Validation

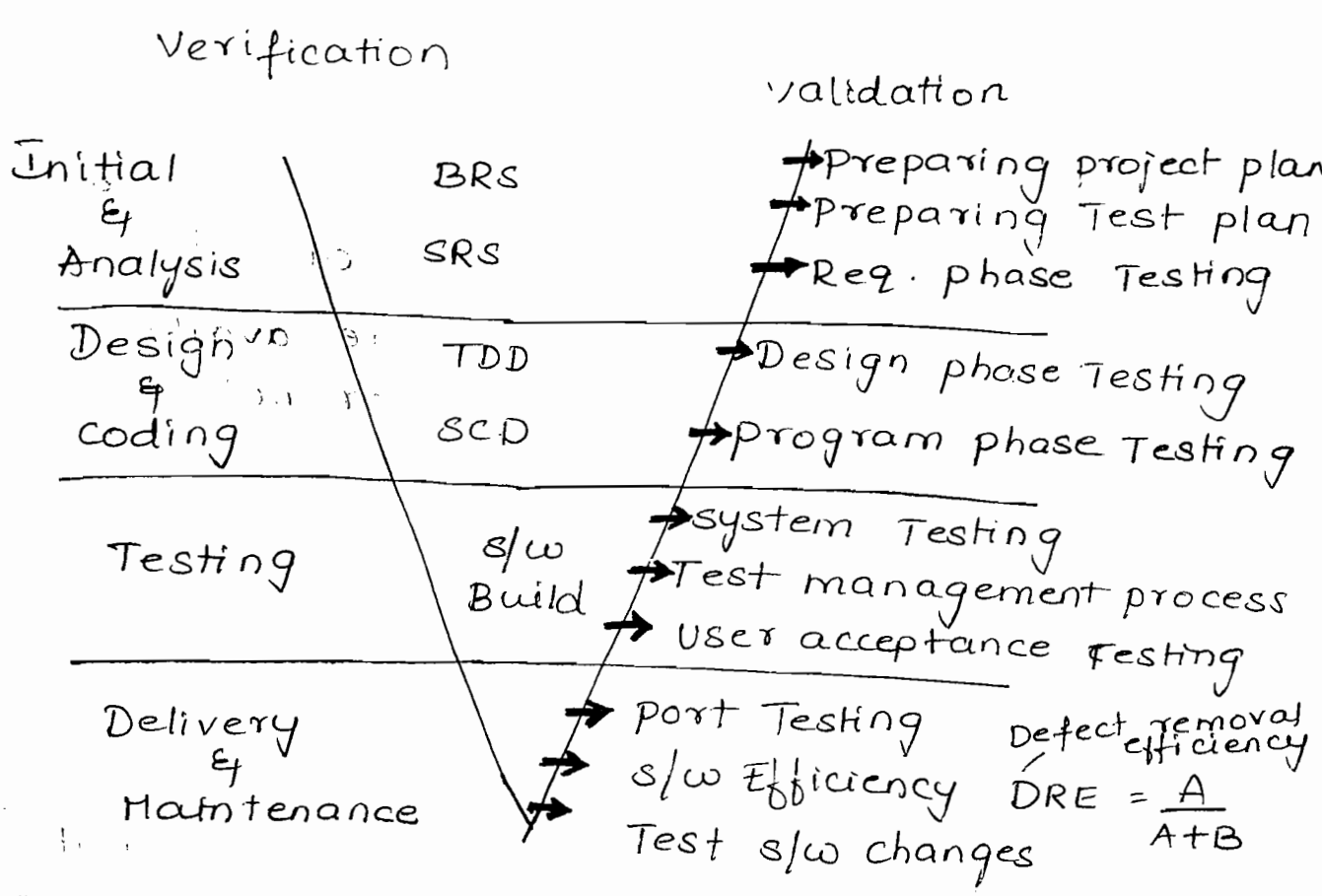
port: final destination which can hold.

Advantages :-

1. As the verification and validation are done and Test management process is followed, the outcome will be a quality product.

Drawbacks :-

1. Time consuming.
2. costly model.



A = Defects found by testing team

B = Defects raised by the customer

Installation Engineers - post
. will do

TYPES OF TESTING

1. Build Acceptance Testing (or) Build verification Testing (or) Sanity Testing:

It is a type of testing in which one will ~~one with~~ conduct overall testing on the released build. Inorder to check whether it is proper for further detailed ~~ch~~ Testing or not.

Usually, In this type of Testing, they will check the foll:-

1) whether the build can be properly installed or not.

2) whether one can navigate to all the pages of the application or not.

3) whether all the objects are available and properly arranged or not.

4) whether the reqd. connections are propely established or not.

Note :- Some companies even call this type of testing as smoke testing. But some companies say that soon after the build is developed, just before releasing that to the testing Dept., the development people will check the overall properness of the build. That is known as smoke testing. And once it is released to the testing dept., whatever the testers are doing is know as BVT, BAT or sanity testing.

repititive

2. Regression Testing :-

It is a type of testing in which one will test the already tested functionalities again and again. usually we do it in 2 scenarios :-

1. whenever the testers find some defects, send it to the ~~testi~~ Development Dept., after rectification, developers will release the next build once the

check the defect functionality as well as the related functionality are working fine or not.

2. Whenever some new features are incorporated or added, next build is released to the testing department then the test engineers will once again test all the related features of those new features. Inorder to check whether they are working same as previous or not.

Note :- Testing the new features for the 1st time is known as new testing but not regression testing.

3. Retesting :-

It is a type of testing in which one perform testing on the same functionality again and again with multiple sets of values inorder to come to a conclusion whether it is working fine or not.

Note :-

a) During regression testing also, retesting will be conducted.

b) regression test starts from 2nd build & continues upto last build.

c) Retesting starts from 1st build & continues upto last build.

3) 4. α-Testing :-

It is a type of user-acceptance testing done in the company by our test engineers in the presence of the user.

β-testing :-

It is a type of user-acceptance testing done in the client's place either by the end users or by the third party testing experts.

Static Testing	Dynamic Testing
It is a type of testing in which one will perform testing on the application or its related factors without performing any actions.	It is a type of testing in which one will perform testing on the application or its related factors by doing some actions.
Ex: Document Testing, GUI Testing, code reviews etc.	Ex: Functionality Testing

Note: There are only 2 types of testing.
1. Static 2. Dynamic.

6. Installation Testing :-

It is a type of testing in which one will install the application into the environment by following the guidelines provided in the deployment document inorder to check whether those guidelines are suitable for installing the application into the environment or not.

7. Compatability Testing :-

It is a type of testing in which one will install the application into multiple environments prepared with different combinations inorder to check whether it is suitable with those environments or not.

Note :- It usually this type of testing is conducted to products rather than projects.

8. Monkey Testing :-

It is a type of testing in which one will perform testing on the application by doing some actions intentionally inorder to check the stability of the application.

9. Usability Testing :-

It is a type of testing in which one will check whether the application is user-friendly or not.

10. End-to-End Testing :-

It is a type of testing in which one will perform testing on the end-to-end scenarios of the application.

Ex :- Login
Balance Enquiry 20,000
withdraw 5,000
Balance Enquiry 15,000
Logout.

5|1

11. Exploratory Testing :-

It is a type of testing in which the domain experts will perform testing on the application without having the knowledge of requirements just by exploring the functionality.

* for product.

12. post Testing :-

It is a type of testing in which one will install the application into original client's environment and check whether it is suitable with that environment or not.

(product or project)

13. Security Testing :-

It is a type of testing in which one will check whether the application properly protected or not. To do the same, use black box tester concentrate on the following areas

a) Authentication Testing

↳ uniform resource locator.

c) Firewall leakage testing

a) Authentication testing:

It is a type of testing in which one will try to enter different combinations of username and passwords and check whether the application is allowing only the authorized users or not.

b) Direct url Testing:

It is a type of testing in which one will ~~try~~ to enter direct url's of secured pages and check whether the application is allowing those pages or not.

c) Firewall leakage Testing:

It is a type of testing in which one will enter into the application as one level of user and try to access the application beyond his limits inorder to check whether the application is allowing him to access or not.

14. Soak Testing (or) Reliability Testing:-

It is a type of testing in which one perform testing on the application continuously for long period of time inorder to check the stability of the application.

(Type of stress testing)

Mutation Testing :-

Mutation testing is a generic type of testing in which one will perform testing on the application or its related factors by doing some changes to them.

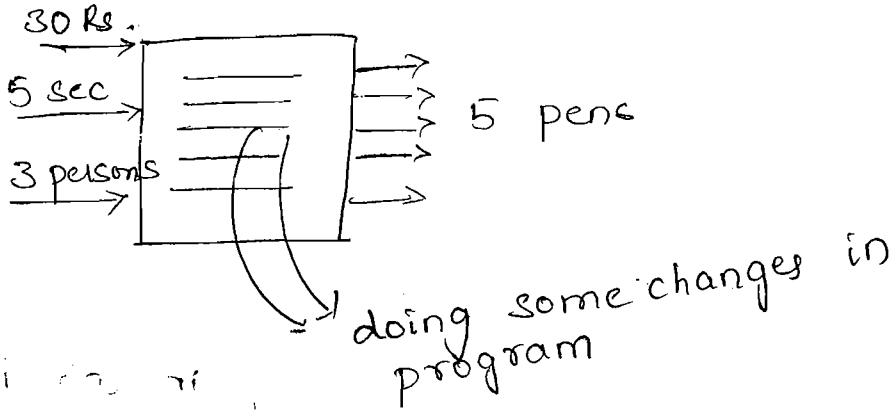
Ex: Regression testing, compatability testing

Ad-hoc testing :-

(own, temporary)

It is a type of testing in which one will check the application in their own style after understanding the requirements clearly.

Note: usually this type of testing is encouraged finally after the formal testing is finished.



SOFTWARE TESTING LIFE CYCLE

STLC contains the foll. phases :-

1. Test planning
2. Test Development
3. Test Execution
4. Result Analysis
5. Bug tracking
6. Reporting.

1. Test planning :-

Plan :- plan is a strategic document which describes how to perform a task in an effective, efficient and optimized way.

Optimization :- optimization is a process of utilizing the available input resources to their level best and getting the maximum possible output.

Test plan :- Test plan is a strategic doc. which contains some information, that describes how to perform testing on an application in an effective, efficient and optimized way.

*** Test plan will be prepared by the Test Lead.

6|1 . Index / contents of the Test plan

1.0 Introduction

1.1. objective

The purpose of the document will be clearly mentioned here in this section.

1.2 Reference Documents

The list of all the documents that ar referred to prepare the test plan will be listed out here in this section.

Ex: project plan
SRS

2.0. coverage of Testing

2.1 Features to be tested

The list of all the features that are in the scope and planned for testing will be listed out here in this section.

2.2 Features not to be tested

The list of all the features that are not planned for testing are mentioned here in this section.

(a) Features that are out of scope
(b) Low risk features
(c) Features that are planned to be incorporated in future.
(d) Features that are skipped based on the time constraints

---

project
less time for completion
ask for 1 2 3 → more no. of hrs.

3.0 Test strategy

Test strategy is an organisational level plan which is used for testing all the projects in that organisation.

Test plan

Test plan is a project level term which is used for testing a particular project.

Note :- Test strategy will be common for all the projects but Test plan will be individual for every project.

Note : There may be slight changes in the test strategy also upon customer's request.

3.1 Levels of Testing

The list of all the levels of testing that are maintained by that company will be mentioned here in this section

3.2 Types of Testing

The list of all the types of testing that are maintained by that company will be mentioned here in this section.

3.3 Test Design Techniques

The list of all the techniques that are maintained by that company will be ~~liste~~ mentioned here in this section.

---

*Test strategy & plan Difference

Technology ↘

Boundary value Analysis (BVA)
Equivalence class partition (ECP)

3.4. configuration management

To be discussed.

3.5 Test metrics

clear measurement

The list of all the things that are planned to be measured during the testing process in that company will be mentioned here in this section.

3.6. Terminology

The list of all the terms and the corresponding meanings used in that company will be maintained here in this sectio

3.7 Automation plan

The list of all the areas that are planned for automation will be mentioned here in this section.

3.8 List of Automated tools

The list of all the automated tools that are used in that company will be listed out here in this section.

FATAL	sev 1	S 1	1	} Terminology will be diff. from company to
MAJOR	sev 2	S 2	1	

7/1

4.0 Base criteria (condition)

4.1. Acceptance criteria

When to stop testing will be clearly described here in this section.

Ex :- Whenever the user acceptance testing is passed, Then stop the testing

Ex :- stop the testing at the deadlines of the project.

4.2 Suspension criteria

When to suspend the build will be clearly specified here in this section.

Ex :- Whenever so many pages of the application are not available

Ex : whenever the application cannot be installed into the environment.

5.0 Test Deliverables

The list of all the documents that are to be prepared during the testing process will be listed out here in this section.

Ex :- Test case Doc., review report Doc., Defect profile Doc.

6.0 Test Environment

The complete details of the environment that is about to be used for testing purpose will be clearly described here in this section.

7.0 Resource planning

Who has to do what will be clearly described here in this section.

The starting dates and the ending dates of each and every task will be clearly planned and specified here in this section.

9.0 Staffing and Training :-

To accomplish the project successfully, if an staff need to be recruited or training need to be provided, then that details will be clearly specified here in this section.

10.0 Risks and contingencies

The list of all the potential risks and the corresponding solution plans will be specified here in this section.

Ex: (Risks)
1. Employees may leave the organisation in the middle of the project.
2. unable to deliver the project within the deadlines.
3. unable to test all the features within the given time.
4. customer may impose (↗ bring forward) the deadlines.

(Contingencies)

Ex:
1. Employees need to be maintained on the bench.
2. proper plan ensurance.
3. ~~unable to test all the features in given time~~ & priority based headlines
4. what to be skipped need to checked in case of imposed deadlines.

11.0 Assumptions

The list of all the assumptions that are to be assumed by the testing people will be listed out here in this section.

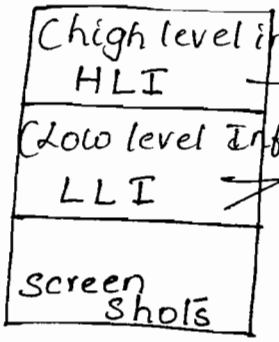
12.0 Approval Information

Who has approved and when it is approved will be clearly mentioned here in this section.

Test Development

↳ Test cases Development

FRS



8.1 use case :-

usecase is a description of functionality of certain feature of an application interms of actors, actions and responses.

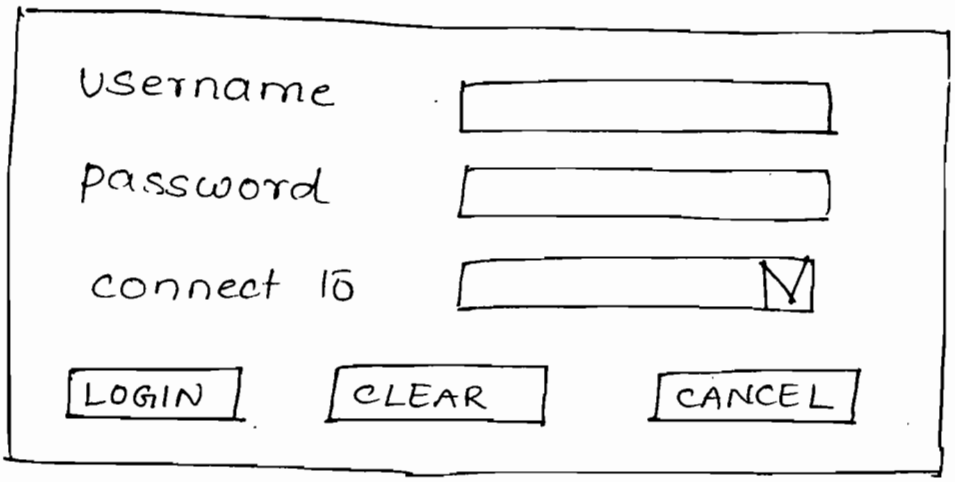
---

usecase - explain Functionality ↑

behaviour of appl.

Input information reqd. for preparing the usecases:

1. Snapshot



point a user can perform action in appl.
↓
fnl pt.

Functional requirements

1. Login screen should contain username, password, connect to fields, Login, clear and cancel buttons.
2. connect to field shouldnot be a mandatory field but it should allow the user to select a database option whenever he requires.
3. upon entering valid username, password and clicking on login button, corresponding page must be displayed.
4. upon clicking on clear button, all the fields must be cleared and the cursor must be available in the username field.

depends on normal or administrative user.

Customers requirements ⟶ functional req's.

↙

special req's, validations, Business rules

5. upon clicking on cancel button login screen must be closed.

Special req's (or) validations (or) Business rules

1. Initially whenever the login screen is invoked (opened), the login and clear buttons must be disabled.
2. ~~clear~~ cancel button must be always enabled.
3. upon entering some information into any of the fields, then the clear button must be enabled.
4. upon entering some information into username and password fields the login button must be enabled.
5. Tabing order must be username, password, connect to, Login, clear and cancel.

Usecase Template

1) usecase name :
2) Breif Description of usecase :
3) Actors involved :
4) special requirements :
5) pre conditions :
6) post conditions :
7) Flow of events :

Explicit — given by ~~reqs~~ customer
Implicit — not specied by customer by analysed by

Usecase Name : Login usecase.

Breif Description of usecase : This usecase describes the functionality of all the features of the Login screen.

Actors involved : Normal user
Admin user

special requirements :

Explicit req's :

1. Initially whenever the login screen is opened, the login & clear buttons must be disabled
2. cancel button must be always enabled.
3. upon entering some info. into any of the fields then the clear button must be enabled.
4. upon entering some info. into UN & pwd fields the login button must be enabled.
5. Tabing order must be UN, pwd, connect to, Login, clear & cancel

Implicit req's :

1. Initially the cursor must be available in the UN field whenever login screen is invoked.
2. whenever invalid UN & valid pwd is entered & click LOGIN message must be displayed like "Invalid UN Please Try again".
3. upon ~~click~~ entering valid UN & Invalid pwd, and clicking on LOGIN button, the foll. message must be displayed " INVALID PWD PLZ TRY AGAIN"

---

special req's are of 2 types.

1. Explicit req's

The special req's given by the customer are known as explicit req's.

2. Implicit req's.

The req's that are analysed by the business analyst which will increase the value of the application are known as implicit req's.

4. upon entering both the UN & pwd invalid and clicking on LOGIN button, the foll. message must be displayed "Invalid UN & pwd please Try again"

pre conditions : Login screen must be available.

post conditions : Either home (or) admin page for valid users and error message for invalid users.

Flow of events :

Main flow :

Sl NO	Action	Response
1.	Actor invokes the application.	Application displays the login screen with the foll. feilds- UN, pwd, connect to, Login, clear and cancel.
2.	Actor enters valid UN, valid pwd and clicks on Login button.	Authenticates application, displays either home page or admin page depending upon the actor entered.
3.	Actor enters valid UN, valid pwd, select a database option and clicks on Login button.	Authenticates, application displays either home page or admin page depending upon the actor entered along with selected database connection.
4.	Actor enters invalid UN, valid pwd and clicks on login button.	Goto alternative flow table 1.

5.	Actor enters valid UN, invalid pwd and clicks on Login button	Goto alternative flow table 2.
6.	Actor enters both un, pwd invalid and clicks on Login button.	Goto alternative flow table 3.
7.	Actor enter some information into any of the fields and clicks on clear button.	Goto alternative flow table 1.
8.	Actor clicks on cancel button.	Goto alternative flow table 5

Alternative flow:
Table 1 (Invalid username)

	ACTION	Response
	~~Invalid username~~	
1.	Actor enter invalid username, valid password and clicks on Login button.	Authenticates, application displays the following error message "Invalid username, please try again"

Authentication: It checking valid UN, pwd & displaying desired page.

Alternative flow Table 2 :
(Invalid pwd)

	Action	Response
1.	Actor enters valid UN and invalid password and clicks on Login button.	Authenticales, application displays an error message "Invalid pwd, please try again"

Alternative flow Table 3
(Invalid UN & Pwd)

	Action	Response
1.	Actor enters Invalid username and invalid password and clicks on Login button.	Authenticales, appl. displays error message "Invalid UN, Invalid pwd, plz try again"

Alternative flow Table 4.
(Clear click)

Action	Response
Actor enters some info. into any of the fields and clicks on ~~Login~~ clear button.	All the feilds in login screen are cleared & cursor is placed ~~available~~ in UN field.

Alternative flow table 3:
(cancel click)

Action	Response
Actor clicks on cancel button	Login screen is closed

9/1. Identify, understand, document.

** Guidelines to be followed by a test engineer once the usecase document is recieved

1. Identify the module to which the usecase belongs to.
   Ans: Security module
2. Identify the functionality of the usecase wrt the total functionality.
   Ans. Authentication
3. Identify the functional points and prepare the functional point document.
4. Identify the inputs reqd. to perform testing.
   Ans: Valid & Invalid UN's and pwd's.
5. Identify the actors involved in the usecase.
   Ans: Normal user & Admin user.
6. Identify whether the usecase is linked with any other usecase.
   Ans: Linked with home page (or) Admin page

7. Identify the pre conditions

Ans: Login screen must be available.

8. Identify the post conditions.

Ans: Either home page or admin page for valid users and error message for invalid user.

9. understand the main flow of the application.

10. understand the alternative flow of the application.

11. understand the special requirements.

12. Document the testcases for the main flow.

13. Document the testcases for the alternative flow.

14. Document the testcases for special req's.

15. prepare the cross reference matrix (or) traceability matrix.

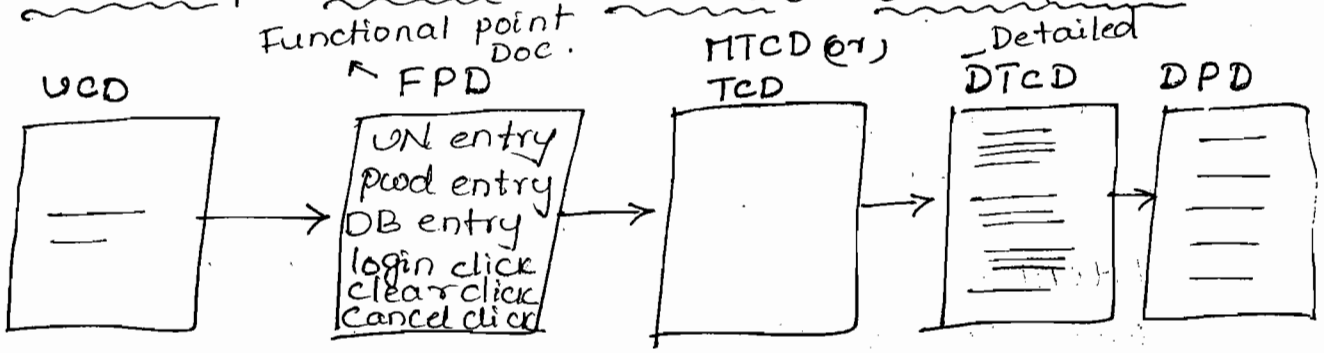
↓

combining different points

Functional point :

The point where a user can perform some action is known as functional point.

TESTING PROCESS RELATED DOCUMENTS



Traceability matrix (or) cross reference matrix

It is a document which contains a table of linking information used for tracing back for the reference in any kind of confused or questionable situations.

Ex: 1) complete Traceability Matrix (CTM)

UCD	FPD	TSD	TCD	DPD
9.2	8	3	28	1
20	19	4	42	2
21.3	28	21	108	3
28.4	—	—	128	4 — Look & feel Defects

2) Requirement Traceability Matrix (RTM)

TC ID	Req. ID
1	1
2	1
3	1
4	1
5	1.2
6	1.2
7	2
8	2.3
9	2.4

3) Defect Traceability matrix (DTM)

DID	TCID
1	28
2	42
3	102
4	109
5	121

Types of Testcases :

Testcase are broadly divided into 3 types.

1. GUI testcases
2. Functional testcases
3. Non-Functional testcases

load
performance
compatability

Functional Testcases are further divided into 2 types.

1. positive testcases
2. Negative testcases

Guidelines for preparing GUI testcases :

1. check for the availability of all the objects.
2. check for the consistency of all the objects.
3. check for the alignments of the objects if customer requirements are given.
4. check for the spellings and grammar.

if we can test just by looking and feeling, that will fall under GUI testcases.

Guidelines for writing the positive testcases:

1. A Test engineer should have positive mind set.
2. He should consider the positive flow of the application.
3. He should use the valid inputs from the point of functionality.

Guidelines for writing the negative testcases:-

1. A test engineer should have -ve mindset.
2. He should consider the -ve flow of Appln.
3. He should use minimum 1 invalid input for each set of data.

10|1

Testcase Template:

1 Project Name :
2 Module Name :
3 prepared by :
4 prepared date :
5 Reviewed by :
6 Review date :
7 Approved by :
8 Approved date :

9 Test objective :

10 Test Scenario :

11. Test Data :

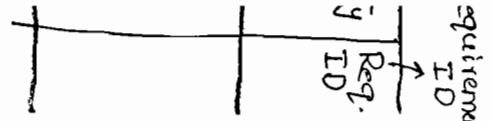
12 Test cases :

Testcase ID	Tc Type	Pre-Requisite	Desc or Test steps	TD (Test Data)	EV	AV	Result	BNO (Build No.)	Priority	Req. ID (Requirement ID)

→ Testcase must be correct as well as good

→ " should be simple

→ " " " clearly understandable



Testcase ID	Testcase Type	Prerequisite	Description/Teststeps	Test Data	Expected value	Actual value	Result	BNO	priority	Req ID
1	+ve	-NA-	Invoke the application	-	Login screen must be displayed with the following fields: UN/pwd, connect to Login, clear & cancel.					
2.	GUI	-NA	check for availability of all the objects as per the Login objects table	LOT	All the obj. must be available as per the login obj. table.					
3	GUI		check ~~whether~~ for the *Consistency of all the objects		All the obj. must be consistent with each other.					
4.	GUI		check for the spellings of all the objects.		All the obj. must be spelled properly.					
5	GUI		check for initial position of the cursor.		Initially cursor must be positioned in the UN field.					
6.	GUI		check for the enabled prop. of Login, clear and cancel buttons.		Initially Login & clear buttons must be disabled & cancel button must be enabled	L, clear & cancel ~~are~~ buttons are enabled	Failed			

ns	Testcase type	Pre-requisite	Description	Test Data	~~Actual~~ Expected value	Actual value	Result	B NO	Priority	Req. ID.
e7	+ve		Enter some information into any of the fields & check for enabled prop. of clear button.		clear button must be enabled.					
	+ve		Enter some information into both username and password feilds and check for the enabled property of login button.		Login button must be enabled.					
	+ve		Enter username and password as per the VIT and click on login button	VII	corresponding pages must be displayed as per the VIT.					
	+ve		Enter the username, pwd as per the VIT, select a database option and click on Login button.	VIT	Corresponding pages must be displayed as per the VIT with the mentioned database connection					
			Enter some information into		All the fields	All fields				

				base connection				
11.	+Ve	Enter some information into any of the fields & ~~cbc~~. click on ~~Cancel~~ Clear button		All the fields must be cleared & The cursor must be placed in The username field.	All fields are cleared but & cursor is not displayed in UN field	Fail		
12	+Ve	click on The cancel button.		Login screen must be closed.				
13	+Ve	check for The tabing order of The objects.		Tabing order must be as foll: UN, Pwd, connect to, clear, cancel				
14	-Ve	Enter UN, pwd as per IVIT and click on Login button	IVIT	corresponding error messages must be displayed as per IVIT.	corresponding errormgs are not displayed as per IVIT	Fail		
15.	-Ve	Enter some information only into the UN field and check for The enabled prop. of Login button.		Login button must be disabled.	Login button is enabled	Fail		
16.	-Ve	Enter some information only into pwd field & check of enabled prop. of Login.		Login button must be disabled.	Login button is enabled	Fail		

* Login object Table.

SlNO	obj. Name	obj. Type
1	username	Textbox
2	password	Textbox
3.	connect to	Combo box
4	Login	button
5	cancel	button
6	clear	button.

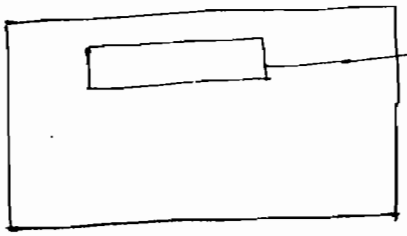
* If reqd. make spelling table

* valid inputs table (VIT)

Sl.NO	username	password	Expected pag	Actual page	Result
1	suresh	@TP	Admin		
2	chiru	sridevi	homepage		
3	vamsi	Illu	homepage		
4	Krishna	Radha	homepage		
5	shiva	parvati	home		
6	Admin	Admin	Admin		

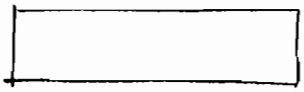
Invalid inputs table (IVIT)

Sl.NO	UN	pwd	Error msg	EV	AV	Result
1	suri	@TP	IUN PTA			
2	chiruta	sridevi	IUN PTA			
3	vamsi	savitri	IPwd PTA			
4	shiva	Ganga	IPwd			



Text box.
min 4 char to
max 20 char.
↓

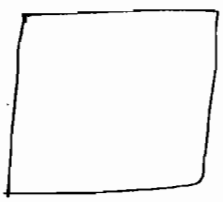
LB-1	LB	LB+1		UB-1		
~~BV~~+1	~~BV~~	~~BV=1~~	mv	~~BV=1~~	~~BV~~	~~B++~~
5	4	3	12	19	20	21

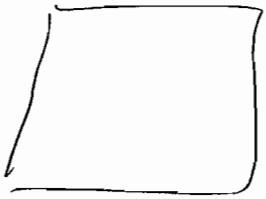
email 

1) @ and _
2) only small alphabels
3) 4 ch to 20 ch.

1. enter capital letters in email textbox — should not accept. (EV)
2. enter small alphabels

1	+ve	enter The values as per VIT	should accept
2	-ve	enter The values as per IVIT	should not accept

VIT 

IVIT 

IVIT

Sl NO	Input
1	abc
2	ABCD
3	1238
4	ABC123
5	5.5
6	
7	ab@_><.$2
8	abcd ABCD <>., 1234 SS4Z
9	abcd abcd abcd abcd abcd z

VIT

Sl.NO	Input

Test Design Techniques

While designing the testcases, if at all the test engineer feels any difficulty, then they will use the techniques inorder to develop the testcases easily.

There are 2 famous techniques used by most of the companies.

1. Boundary value Analysis
2. Equivalence class partition

Boundary value Analysis (BVA)

Whenever one need to develop the test-cases for a range kind of input then it is suggested to use a technique called BVA.

Boundary value Analysis says that instead of testing on whole range, just concentrate on the boundaries if at all the boundaries are working fine then one can come to a conclusion then the whole range is working fine.

Equivalence class partition (ECP)

Whenever one need to develop the testcases for huge range of data or a mixture of so many requirements then it is suggested to use a technique ECP.

Equivalence class partition says that divide the inputs into different classes and then write the testcases for each class seperately.

Ex:

Develop the testcases for an e-mail textbox whose requirements are as follows:

1. It should accept min. 4 characters, max 20 characters
2. It should accept only small alphabets
3. It should accept @ and _ special characters only.

BVA

LB-1	LB	LB+1	MV	UB-1	UB	UB+1
3ch	4ch	5ch	12ch	19ch	20ch	21ch

Ecp

Valid	Invalid
4 ch	3 ch
5 ch	2 ch
12 ch	A-Z
19 ch	All the remaining special characters apart from @ & _
20 ch	Alphanumeric
a-z	0-9 (decimal no.s)
@	
_	

VIT

Sl NO	Input
1	abcd
2	ab@cd
3	ab__d
4	abcd abcd abcd
5	abcd abcd abcd abcd@_$
6	abcd abcd abcd abcd lmno

IVIT

Sl NO	Input
1	abc
2	ABCD
3	1238
4	ABC123
5	5·5
6	[ ]
7	ab@_><·$3
8	abcd ABCD >< ·, 1234□ SS43
9	abcd abcd abcd abcd abcd3

Testcases :

Testcase Id	Testcase TYPE	Description	Ev
E. 1	+ve	Enter the values into the Textbox as per VIT	Text box should accept
2	-ve	Enter the values into the Textbox as per IVIT	Textbox shouldnot accept

Test Execution

In the execution phase the test engineer will be doing the foll. 1) he will perform the action as it is described in the description ~~column~~ part.

2) He will observe the actual behaviour of the application.

3) He will document the observed information in the actual value column.

Result Analysis:

In this phase, the test engineer will be comparing the expected values with the actual values & if both are matching, he will decide the result as pass otherwise fail.

Note: whenever the testcase can't be executed then the result will be blocked.

Bug tracking

Bug tracking is a process in which all the defects are identified, isolated & managed.

Defect profile template

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)

1 → Defect ID

2 → Testcase ID

3 → Defect Description

4 → steps for reproducibility

5 → submitter

6 → Date of submission

7 → Build NO.

8 → version NO

9 → Assigned to

10 → severity

11 → priority

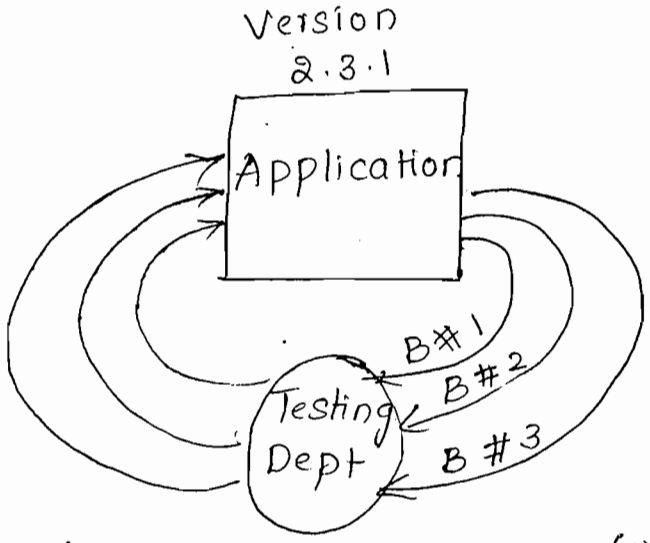
12 → status

Defect ID	Tc ID	Defect Description	Steps for reproducibility	Submitter	DOS	BNO	VNO	AT	Severity	Priority	status
	6	Initially login & clear buttons are enabled instead of being disabled.	— NA — (look & feel)								
	11	upon clicking on clear button, all the fields are cleared but the cursor is not positioned in the username field.	1. Enter some info. into any of the fields. 2. click on clear button. 3. observe that all the fields are cleared but the cursor is not positioned in the UN field.	shanti sri	20/1	1	2.3.1				
	14	upon entering suri as UN, @TP as pwd and clicking on Login button, admin page is displayed instead of error message.	1. Enter suri into the UN field 2. Enter @TP into the pwd field 3. click on Login button. 4. observe that Admin page is displayed instead of error msg.	shanti sri	20/1	1	2.3.1				
	15 & 16	upon entering some info. into UN or pwd fields only, Login button is enabled instead of being disabled.	1. Enter some info. into UN (or) pwd fields. 2. check for the enabled prop. of login button. 3. observe that Login button is enabled instead of being disabled.	shanti sri	20/1	1	2.3.1				

used for reproducing the same defects.

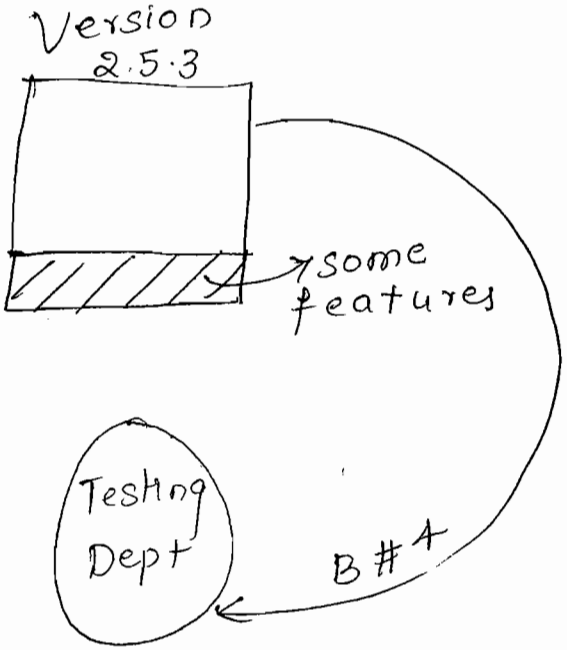
Configuration mgt team will assign

Build NO — **Version NO.**



Whenever App. passes to testing Dept. BNo increases by 1

Whenever some features are added to application. Version no. is changed.



3. Observe that login buttons is enabled instead of being disabled.

being disabled.

Assigned to — Dev lead will fill this column to the corresponding Developer who has developed that particular thing.

Defect ID

The sequence of defect numbers will be specified here in this section.

Testcase ID

The Testcase ID based on which the defect is found will be specified here in this section.

Defect Description

what exactly the defect is will be clearly described here in this section.

Steps for reproducibility

The list of all the steps followed by the test engineer to identify the defect will be specified here in this section.

submitter

The name of the test engineer who has submitted the defect will be specified here in this section.

Date of submission

The date on which the defect is submitted will be specified here in this section.

Build No

The corresponding build no. will be specified here in this section.

Note: Everytime the application is released from the development dept. to the test dept., the build no. will be increased by 1.

Version No.

The corresponding version no. will be specified here in this section.

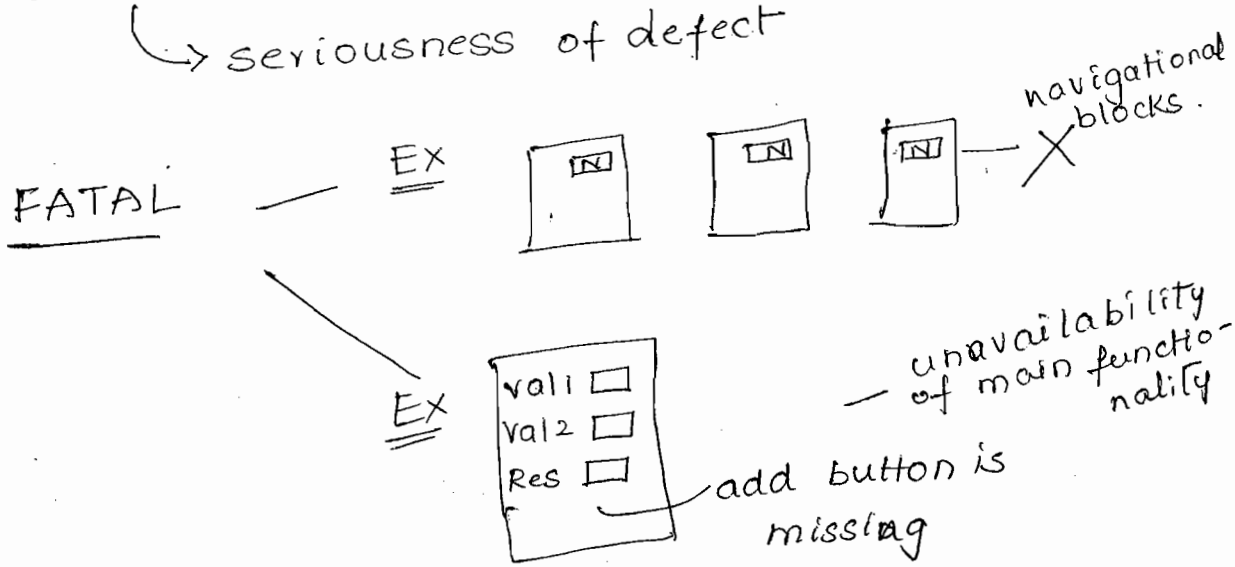
Note: whenever some new features are added to the application then the version no. will be updated by the configuration management team based on the amt. of change made to the application.

Assigned To

The corresponding developers name will be specified by the development lead here in this section for whom the defect is assigned.

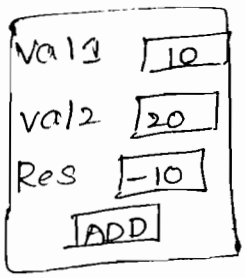
SEVERITY

→ seriousness of defect



Major

Ex:



Major
Functionality is not working as expected

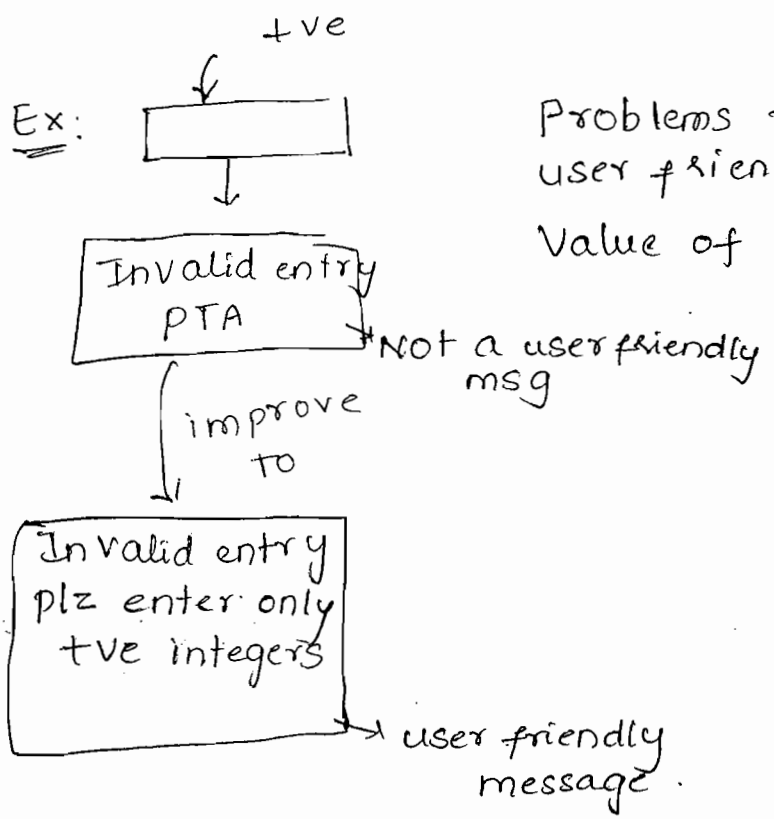
Minor

Ex

No consistency, look & feel prob

Suggestions

Ex:



Problems related to user friendliness, Value of application

Severity describes the seriousness of the defects.

Severity is classified into 4 types

Fatal	sev 1	S1	1
Major	sev 2	S2	2
Minor	sev 3	S3	3
suggestions	sev 4	S4	4

Different Terminologies

Fatal : If at all the problems are related to the navigational blocks or unavailability of major functionality then such type of defects are treated as fatal defects.

Major : If at all the major functionality is not working according to the expectation then such type of defects are treated as major defects.

Minor : If at all the problems are related to the look and feel of the application then such type of defects are treated as minor defects.

Suggestions : If at all the problems are related to the value of the application then such type of defects are treated as suggestions.

PRIORITY

↳ Importance

Initially Test lead assigns priority but Dev. lead changes acc. to situation.

Test case priority } — Test engineers.

Defect priority } — Dev. Department.

priority describes the sequence in which defects need to be rectified.

priority is classified into 4 types.

critical	pri 1	P1	1
High	pri 2	P2	2
Medium	pri 3	P3	3
Low	pri 4	P4	4

different Terminologies.

usually based on the severity, priority will be given. But depending upon different situations, the priority will be changed by the Dev. lead.

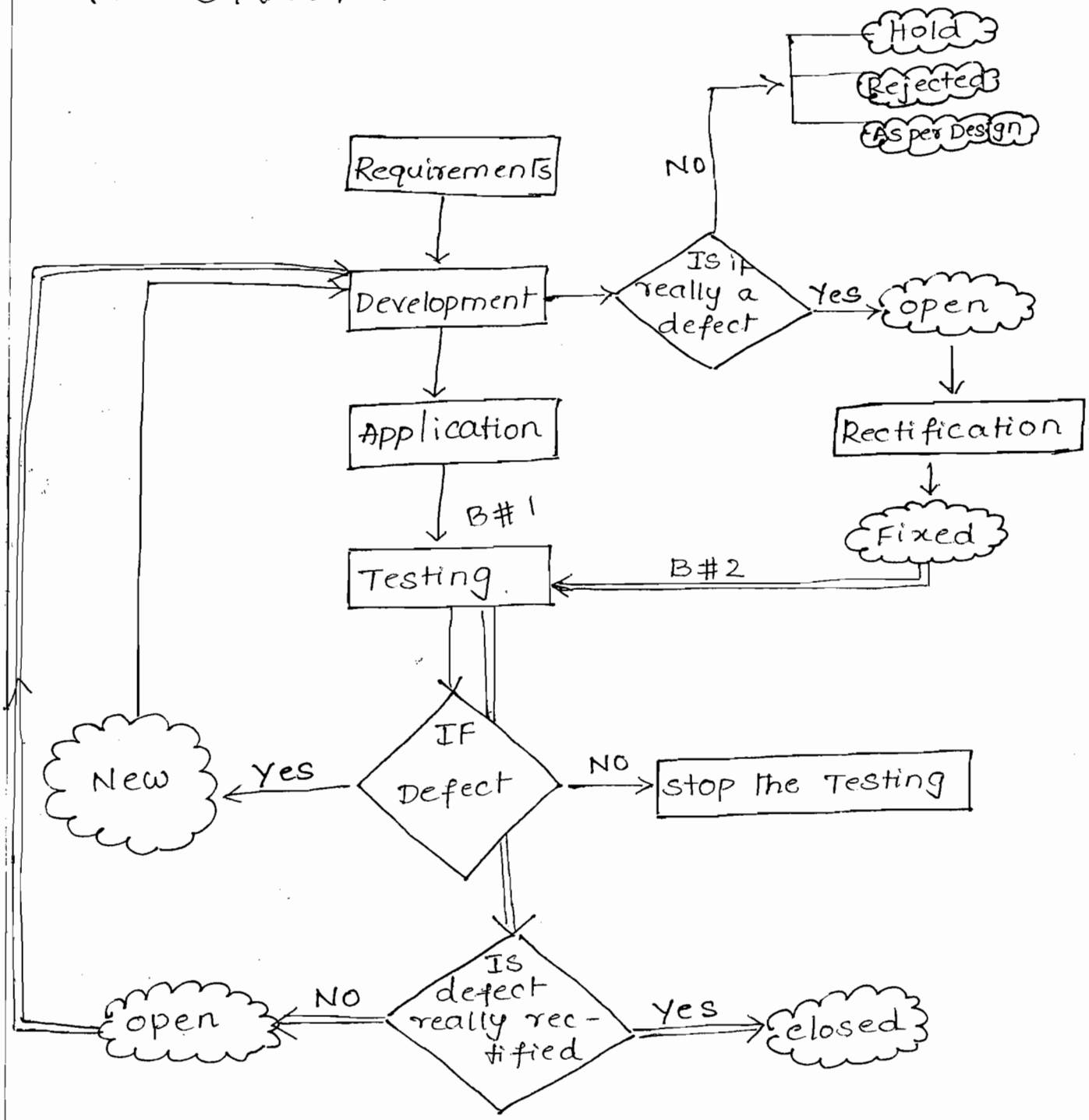
Ex: Least severity & Highest priority case.

whenever there is a customer visit, all the look and feel defects will be given highest priority.

highest severity and least priority and

whenever some part of the application is under development, not released to the testing department, then the testers will raise them as fatal defects but the least priority will be given for them.

Bug Life cycle:



1) Triage meeting — hold
Situation

2) Error — prob ~~is~~ x identified in a program/code
3) Defect — Tester ~~finds~~ identifies problem.
4) Bug — Defects accepted by Developer
5) fault — customer finds a prob.

6) * All defects ~~are~~ can not bugs
but every bug is a defect

∞

New :-

Whenever the defect is newly identified for the first time then the test engineer will set the status as new.

Open :-

Whenever the developer accepts the defect then he will set the status as Open.

Fixed :-

Whenever the developer rectifies the defect before releasing the next build, he will set the status as Fixed.

Reopen & closed :-

Whenever the next build is released, the test engineers will check whether the defects are properly rectified or not. If at all

Otherwise as Reopen.

Hold :-

whenever the developer is confused to accept or reject the defect, he will set the status as Hold.

Note : whenever the defect is in hold status, there will be a meeting on that defect and will finally come to a conclusion whether it is defect or not.

If it is a defect, the developer will change the status to open otherwise the test engineer will change the status to closed

Rejected :-

whenever the developer feels it is not atall a defect then he will set the status as Rejected.

Note : whenever the defect is rejected, the test engineer will once again check it. If he feels really it is not a defect then he will set the status as closed. Otherwise Reopen.

As per Design :-

whenever the developer feels the test engineer has raised the defect without knowing the latest req's then he will set the status as APD

Note : whenever defect is in APD status the Testengineer will check the updated req's doc

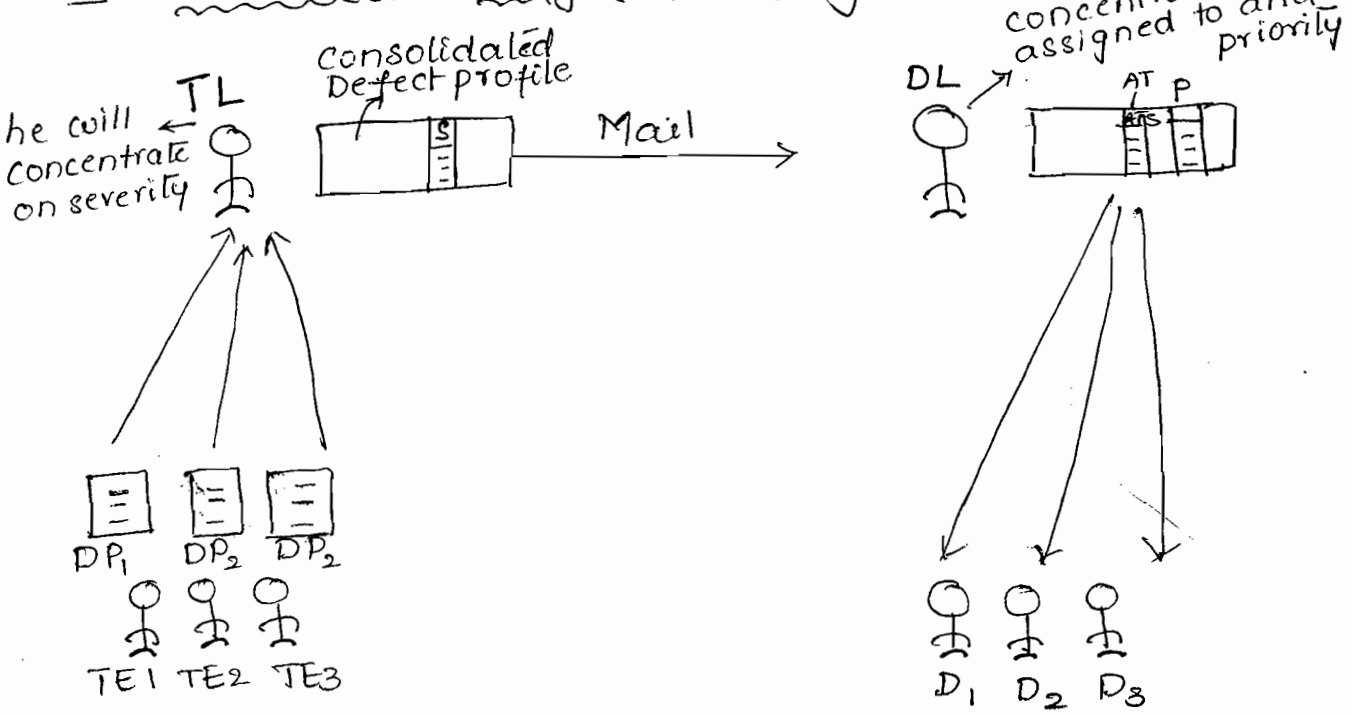
and if he feels it is As per Design, he will set the status as closed otherwise Reopen.

Differed :

whenever the defect is accepted by the developers and if he requires some extra time to rectify that defect, then he will set the status as Differed.

REPORTING :-

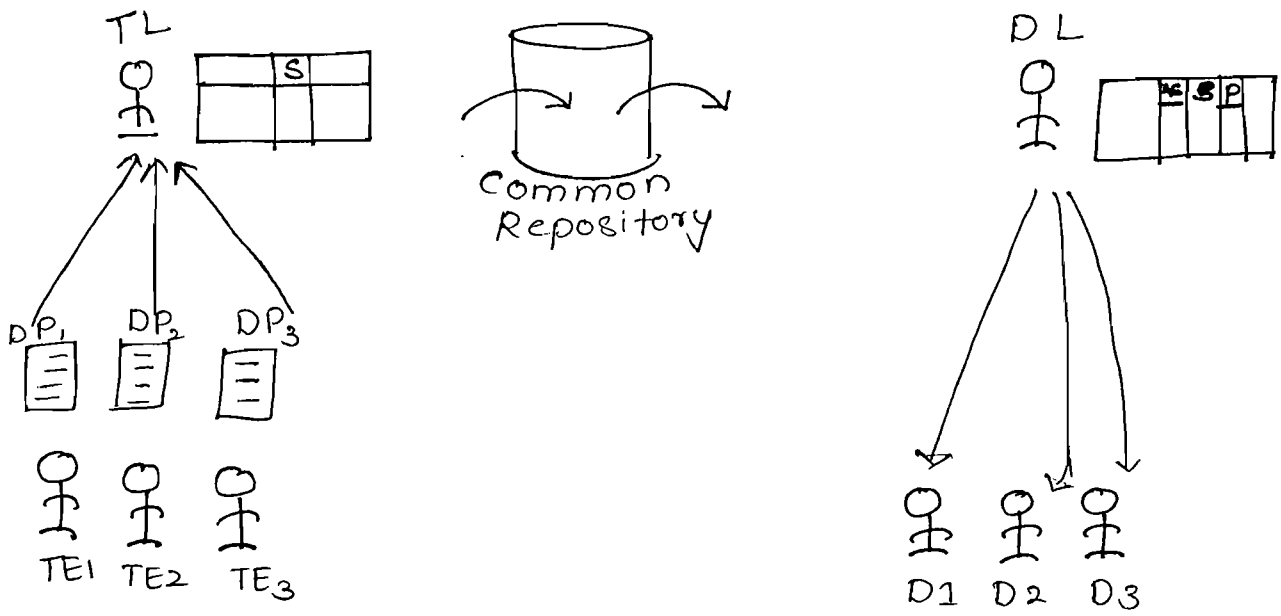
1. Classical Bug reporting process :-



Drawbacks :-

a) Time consuming.
b) No Transperancy.
c) Redundancy.
d) No security

2. Common Repository orienled Bug reporting process



Note :- The Famous common repository managing software are

- Visual source safe (VSS)
- Concurrent version system (CVS)

Drawbacks :

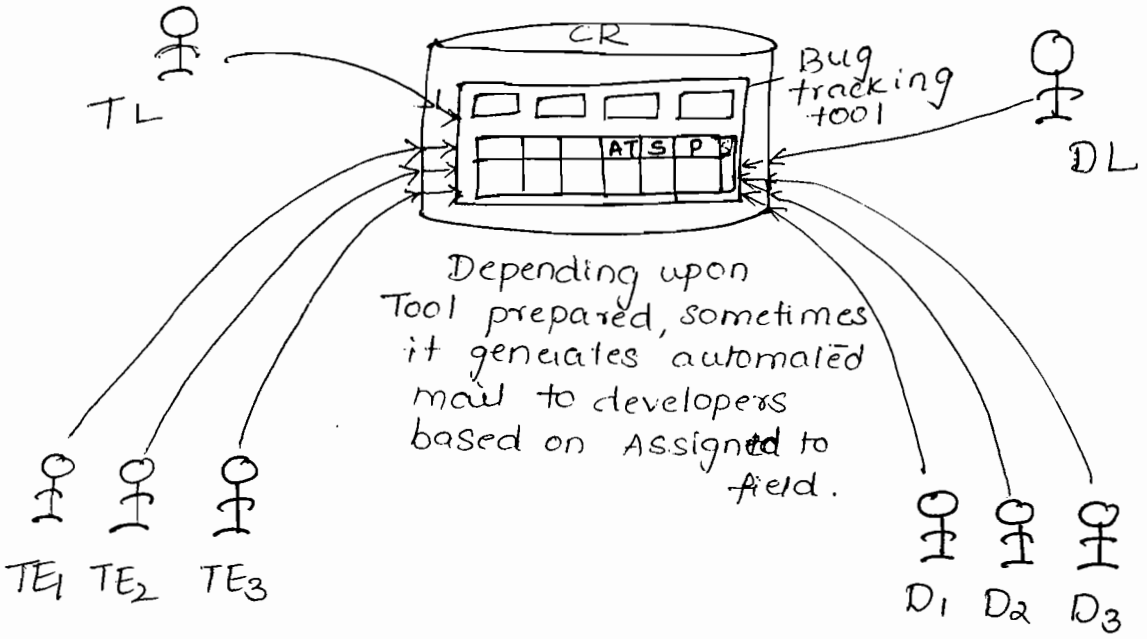
a) Time consuming
b) No Transperancy
c) Redundancy.

3. Bug Tracking Tool orienled Bug Reporting process

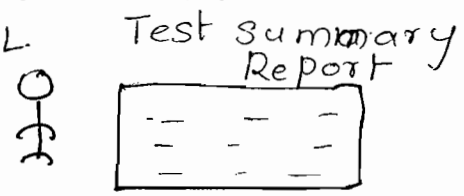
Software application which can be accessed only by the authorized people and provedes all the facilities for Bug Tracking process

Ex:

Bugzilla
PR Tracker problem report.
Issue Tracker etc.



Test closure activity :-



Finally the Test lead will prepare the Test summary Report which contains the following information.

1. No. of Test cases executed in each build
2. No. of cycles of execution.
3. Duration of each cycle.
4. No. of Defects identified in each build etc.

Fe

d
Fe.

RESUME PREPARATION

Header ↗

Name;

Ph-NO:
E-mail ;

Objective : — 2 lines —

Related to person
Relavent to Testing.

summary : Need to write it in bullets.

- Experience
  (3+) → Start with Total experience

  3+ years of experience as a software Tester.
- Involved in testing client-server and web applications.
- Good exposure to SDLC, STLC and BLC.
- Involved in different types of testing like BAT, Regression Testing, Retesting, End-to-End Testing & Ad-hoc testing.
- Good experience in writing & executing The testcases.
- Good experience in testing The functionality of application with the support of Automated tools QTP & QC.
- Good Knowledge of Automaled Tools lik LR.
- comfortable in developing The testscripts using vbscript language

Strengths :- 5 strengths -

1. Quality oriented mind set up.
2. Test-to-break attitude.
3. creativity.
4. Good communication & drafting skills.
5. Good judgement skills.

Don't compromise with Quality

find defect in already tested product.

one can clearly understand Tc's prepared by me.

Skill set (or) Technical skills :-

Software Testing : Manual and Automation.
Automated Tools : QTP, QC & Load runner.
Programing Languages : C
Scripting Languages : Vbscript
Operating systems : windows
RDBMS : SQL Server

Educational profile :

B.Tech (Mechanical)

Working profile :-

1. working as Software Test engineer for — company from Dt. to till date.

2. worked as Test Engineer for — company from Dt. to Dt.

Project profile :

Project Id :
project name :
client :
Duration :
Environment : 1) Browser name, webserver name / 2) appl. server name, 3) Dbserver name, 4) operating system name, 5) main Technology.
Role : as a Tester
Description :
Responsibilities : Involved in
ex: preparing Tc's,
peer reviews

Don't specify Team size.
if at all ques comes — 9

Personal profile :

Date of Birth :
Gender :
Marital status :
Passport Number :
permanent Address :
Mailing Address :

Achievements :

22/12/08

Ways of Testing :-

There are 2 ways of Testing.

1. Manual Testing
2. Automation Testing.

Manual Testing :-

Manual Testing is a process in which all the phases of software testing Life cycle like Test planning, Test development, Test Execution, Result Analysis, Bug Tracking and Reporting are accomplished successfully with human efforts.

Drawbacks of Manual Testing :-

1. More number of human resources are reqd.
2. Time consuming
3. No accuracy
4. Tiredness
5. simultaneous actions are almost impossible.
6. cannot repeat the same task with the same interest.

Automation Testing :-

Automation testing is a process in which all the drawbacks of manual testing are addressed properly and provides speed and accuracy for the existing testing process.

(overcome / nullified)

Drawbacks of Automation :-

1. More expensive (or) Tools r costly.
2. All the areas of application cannot be tested with automated tools.
3. Lack of Automation testing experts in the market.

Note :-

1. Automation Testing is not a replacement for manual testing, it is just a continuation for manual testing.

2. unless and until the application comes to a stable state, it is not suggested to perform automation testing.

Automaled Tool :-

Automaled tool is defined as an assistant of a test engineer which works based on the instructions and information given by the test engineers.

General framework to learn any automaled tool :-

To use any automaled tool, a test engineer should learn the toll :-

1. How to give the Instructions.
2. How to give the Information
3. How to use its recording facility.
4. How to use its playback facility.
5. How to analyze the results.

Types of ~~Au~~ Black box testing Automaled tools :-

1. Functional tools (QTP)
2. Performance tools (Load Runner)
3. Management tool (Quality center)

(Not a testing tool)

QTP

↓

QUICK TEST PROFESSIONAL

Introduction :-

1. Type of the tool :- Functional Tool.
2. Company :- Introduced by Mercury Interactive Incorporation (2002) & Taken over by HP.
3. Scripting Language :- Vbscript.
   [QTP also supports other scripting languages with some internal configuration]
4. Version :-

2002

- 5.5
- 6.5
- 7.0
- 7.6

} same look & feel

- 8.0
- 8.2

}

- 9.0
- 9.1
- (9.2)

} matured version

- 9.5 — latest version called HP's QTP.

ANATOMY OF QTP

↓ overall view

Add-in manager :

1) It is a feature provided by QTP which will appear while opening the QTP with the list of all the available add-ins. 2) The user will select the desired environments and will enter into QTP.
3) Meanwhile the Add-in manager will make the QTP compatable with selected environments.

4) By default QTP will be always compatable with standard windows environment.

5) Whenever we purchase the License of QTP, 3 built-in add-ins will be provided.

1. Active x
2. Visual Basic
3. web.

6) Apart from these Add-ins if any other add-in is reqd., we need to purchase it by paying the extra cost.

Parts of QTP :-

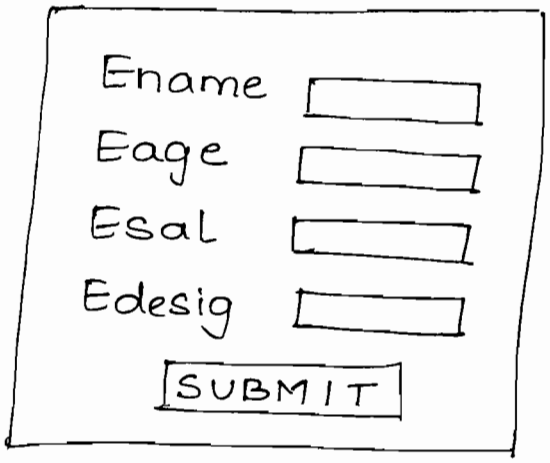
QTP contains 7 parts.

1. Test pane (area)
2. Active screen
3. Data table
4. Debug viewer pane
5. Information pane
6. Missing resources pane
7. Tool options

Test Pane :-

Vbscript for vb applications

EMP



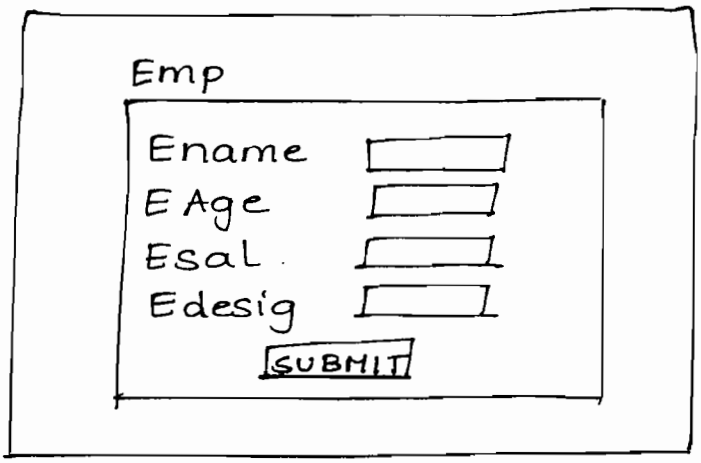

```
Vbwindow ("EMP"). VbEdit ("Ename"). ~~Vb~~set "shantisri"
Vbwindow ("EMP"). VbEdit ("Eage"). set "27"
Vbwindow ("EMP"). VbEdit ("Esal"). set "20,000"
Vbwindow ("EMP"). VbEdit ("Edesig"). set "TestEng."
Vbwindow ("EMP"). ~~VbEdit~~ VbButton ("SUBMIT"). click
```

Vbscript for web applications :-

```
window ("EMP"). winEdit ("Ename"). set "shantisri"
window ("EMP"). winEdit ("Eage"). set "27"
window ("EMP"). winEdit ("Esal"). set "20000"
window ("EMP"). win~~desig~~Edit ("Edesig"). set "TestEng"
window ("EMP"). winButton ("submit"). click
```

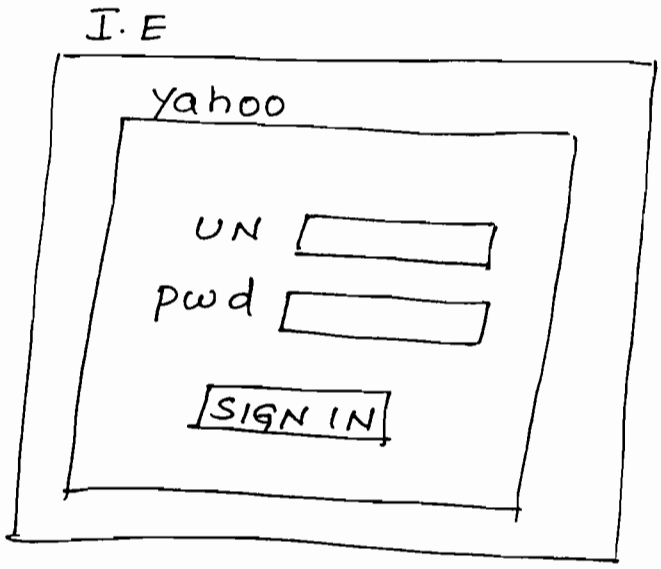
VDScript for web applications :-

I.E



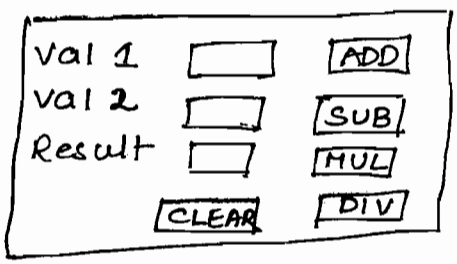

```
Browser("I.E").Page("Emp").WebEdit("Ename").
                                    set "shantisri"
Browser("I.E").page("Emp").WebEdit("EAge").
                                    set "27"
Browser("I.E").page("Emp").WebEdit("Esal").
                                    set "20,000"
Browser("I.E").page("Emp").WebEdit("Edesig").
                                    set "Test Eng"
Browser("I.E").page("Emp").WebButton("SUBMIT")
                                    click.
```

Vb script for web application




```
Browser("I.E").page("yahoo").webEdit("UN").
                                    set "shanti sri"

Browser("I.E").page("yahoo").webEdit("pwd").
                                    set "QTP"

Browser("I.E").page("yahoo").webButton("sign in").
                                    click
```

Vbscript for calculator




```
Vbwindow("cal").VbEdit("val1").set "10"
Vbwindow("cal").VbEdit("val2").set "20"
Vbwindow("cal").VbButton("ADD").click
Vbwindow("cal").VbButton("SUB").click
Vbwindow("cal").VbButton("MUL").click
Vbwindow("cal").VbButton("DIV").click
```

Test pane :-

Test pane is an area provided by QTP which is used for veiwing, developing and modifying The test script.

It shows The script in 2 veiws.

1. Expert veiw
2. Keyword veiw.

Expert veiw :

It shows The saript in Vbscript format.

Keyword veiw :

It shows The script in a graphical user Interface which is further divided into 4 parts.

a) Item
b) operation
c) value
d) Documentation.

Note :- Any modifications done in one veiw will be reflected in The other veiw.

Active screen :- Active screen is an area provided by QTP which holds The snapshots related To each and every script statement and used for The foll :

1) For easily understanding The script
2) For easily enhancing The script.

RECORDING AND RUNNING

Operational overview of Recording :-

During recording, QTP will be doing the following :-

1. It will generate the corresponding testscript statement for every user action.
2. It will store the reqd. related object's information in the object repository.

Operational overview of Running :-

During running, QTP will be doing the followed :-

1. First of all, it will read the instruction (script statement) and understands what action to be performed on which object.
2. It will realize that it need to identify that object inorder to perform the action.
3. It will go to the object repository and will search for the information.
4. If at all the information is found using that information, it will try to identify the original object.
5. If at all the object is identified,

Record & Run settings :-

Record and Run settings is a feature provided by QTP which need to be used compulsarily atleast once for every new test inorder to make the QTP understand what exactly it need to do during recording and running.

Navigation :-

Step 1 : Activate the menu item 'Automation'
Step 2 : select the option 'record & run settings'
Step 3 : Select the desired option
Step 4 : click on apply
Step 5 : click on ok

Recording modes :-

There are 3 types of recording modes in QTP.

1. Normal recording mode / standard recording mode / context sensitive recording mode.
2. Analog recording mode
3. Low level recording mode.

Normal recording mode :-

This recording mode is used for recording the operations performed on the standard gui objects in different situations.

Analog recording

This recording mode is used for recording the continuous operations.

→ Navigation

Keep the tool under normal recording mode.

↳ Activate menu item 'automation'

↳ select the operation, 'Analog recording'

↳ select one of the foll. options.

- (•) record rel. to screen
- ( ) record rel. to foll. window.

* If at all second option is seleded, specify the window title with the help of hand icon.

↓

click on start Analog recording.

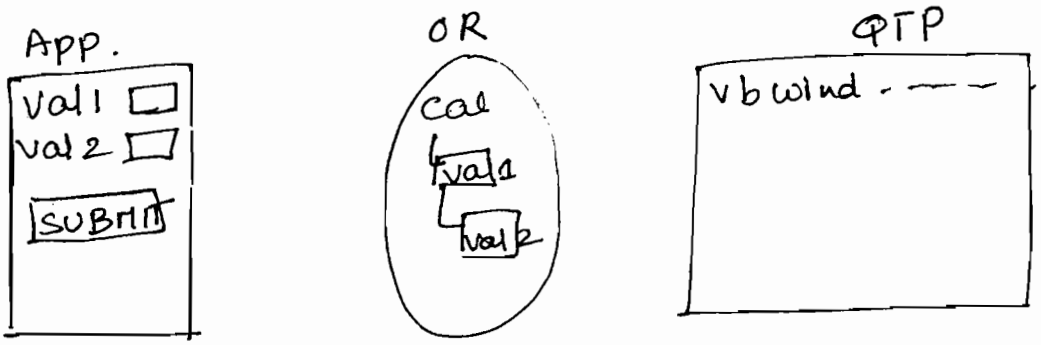
Low-level recording

Low level recording mode is a special recording mode provided by QTP which is used for recording the operations performed on the non supported environments also.

* Note :- usually this recording mode is not much used in the companies because all the operations cannot be recorded, script

Object repository :-

Object repository is a storage place where one can store the object's information and it also acts as an interface between the testscript and the AUT (Application under Test) inorder to identify the objects during execution.



Types of object repository :-

There are 2 types of object repository.

1. Local repository
2. Shared repository

Local repository :-

Every action (purple) in a test will have one local repository. The information present in this can be used only by that action. Local repositories will be automatically created & managed by the QTP itself.

Shared repository :-

One can manually create a shared repository and associate to multiple actions in different tests.

Whenever we feel that some object's information is commonly used by multiple actions then it is always better to create a shared repository and sharing for all those actions.

Advantages :- a) reusability
b) easy to maintain.

Navigation for creating a shared repository:

Activate the menu item 'Resources'
↳ object repository manager
↳ open new file
↳ add reqd. object's info.
↳ save the file in the desired located with .tsr extension.

Navigation for converting the local repository to shared repository:

open object repository
↳ active menu item 'File'
↳ select the option 'Export local objects'
↳ save the file in the desired location with .tsr extension.

Navigation for associating a shared repository to any action :-

Activate the menu item 'Resources'

→ select the option 'Associate repositories'

→ Add the reqd. repositories into the list

→ select the desired repository

→ select the desired action

→ click on associate button [>]

→ click on 'OK'

*** NOTE :- we need to update the shared repositories only in the object repository manager. To do this we need to select an option 'enable editing'.

Operations on Object repository :-

(why How & when)

1. Adding the object's information to the object repository:

Navigation :-

[contd...]

open the object repository
Click on 'Add objects' button

↳ click on title bar of window or on a page or on a desired object

↳ click on ok.

↳ select one of the foll. options.

- (•) selected object only
- ( ) Default object types
- ( ) All object types
- ( ) selected object types.

*** if at all 4th option is selected, Then select the desired type of objects and finally click on ok.

2. Deleting the objects from the object repository

Navigation — Tree structure

Select the desired object in object Heirarchy right click on it, select the option delete

↳ click on yes for confirmation.

3. Renaming the objects:

Navigation

Select the desired object in the object Heirarchy, rt. click on it, Select the option rename, modify the name.

4. Updating the properties list :

Navigation

Select the desired object in the object heirarchy, click on add button (+), Select the desired property to be added, click on ok.

Select the desired properties in the property list,

↳ click on remove button (x)

5. Updating the property's value :-

Select the desired object in the Object heirarcy.

↳ select the desired property in property's list.

↳ click on the configure the value button (<::>), specify the desired value

↳ click on ok.

6. Highlight in application :

It is a feature provided in object repository which is used for highlighting the corresponding object in the application.

7. Locate in repository :

It is a feature provided in object repository which is used for finding the object's information in the object repository.

Object Identification

Object identification concept is completely based on 4 types of properties and an ordinal identifier.

Types of properties :-

1. Mandatory properties
2. Assistive properties
3. Base filter properties
4. optional filter properties.

In case of only Normal identification, QTP will learn the information in the foll. way.

First of all QTP will learn all the mandatory properties at a time. And then it will think whether the mandatory properties are sufficient for identifying the object uniquely. If it feels satisfied then it will stop learning. Otherwise it will learn the first assistive properly and once again think whether all these properties are sufficient for identifying the object uniquely. If it feels sufficient then it will stop learning. Otherwise learns the second assistive prop. and once again thinks. This process continues till the QTP feels sufficient. If at all, all the assistive prop. are learnt and still QTP is not satisfied, then finally, it will learn the ordinal

Note :- The information learnt during the above process will be stored in the Object repository.

Learning process of QTP in case of smart identification also :-

In this case, the learning process will be same as above except in the starting i.e, while learning all the mandatory properties, QTP will also learn all the Base filter props. and optional filter props at a time but stores them in a secret place and will not consider them.

" same as above "

~~Ind~~ Identification process of QTP :-
Smart - web

During execution, QTP will use the information in the foll. way to identify the objects.

First of all QTP will use all the props present in the object repository except ordinal identifier and will try to identify the object. If it fails then it will go to the secret place and will start identifying the object freshly in the foll. way.

First it will consider all the base filter props. and will try to match with all the objects in the application. If at all only one object is matched then that is the object. Otherwise

The objects that are matched with these props will be formed as a list and all the remaining objects will be filtered. Then it will take the support of first optional filter prop. and will try to match with the objects in the list. The objects that are not matched will be filtered and the remaining objects will be present in the list. If at all the list is containing more than one objects, then it will take the support of 2nd OFP and continues the process till the list contains only one object. If at all, all the OFP are used but still if the list is containing more than 1 object, then finally it will check in the object repository whether ordinal identifier is available or not. If it is available, then QTP will use it and roughly identifies the object. Otherwise, QTP cannot identify the object.

Ordinal Identifier :-

There are 3 ordinal identifiers.

1. Location
2. Index
3. creation time.

Location :

Whenever location is selected as an ordinal identifier, QTP will generate some no.'s [illegible] 0 1 2 3 based on the sequence of the

Index:

Whenever Index is selected as an ordinal identifier, QTP will generate some no.'s like 0, 1, 2, 3 based on the sequence of the programs of those objects.

Creation time: (only for Browser object in web environment)

Whenever the creation time is selected as an ordinal identifier QTP will generate some no.'s like 0, 1, 2, 3 based on the loading time of the objects in that browser.

** Object Identification configuration will be *** done in initial stages of project by test lead.

Navigation for object Identification configuration

Activate the menu item 'Tools'
→ select the option 'object identification'
→ select the desired environment
→ select the desired obj. class
→ specify the desired prop. in mandatory & Assistive lists
→ specify an OI
→ If at all smart identification is also reqd. then select the check box
☐ Enable smart identification
→ click on configure

* changes will be reflected from next test.

↳ Specify the desired list of prop. in BF and OF sections

↳ click 'ok'

↳ once again click 'ok'

→ If problem occurs, change configuration only in Object rep. only for that object.

Smart identification :

Whenever the QTP is unable to identify the object with the help of the properties present in the object repository, then the special mechanism provided by the QTP to identify the object smartly by filtering the objects in the application is known as smart identification.

Object Spy :-

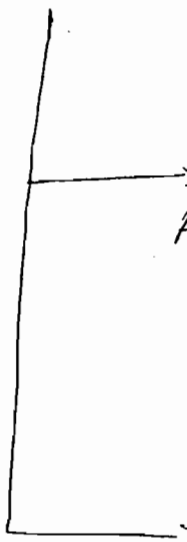
Object spy is a handy feature provided by QTP which shows the complete object information like list of properties and values, list of methods supported by that object, syntax of methods & description of methods then and there immediately of both test objects as well as run time objects.

Navigation

Activate the menu item 'Tools'
→ select the option 'object spy'
→ click on the hand icon
→ specify the desired object in the application.

Objects :- object is something which has some list of properties.

2 Types of objects



→ Run time objects

All the original objects present in the application are known as run time objects.

→ Test objects

Test objects are the reference objects for the original objects created by the QTP and stored in the object repository.

QTP LIFE CYCLE :-

QTP life cycle contains 6 phases.

1. Test planning
2. Generating the basic test
3. Enhancing the test
4. Debugging the test
5. Executing The test
6. Analysing The results

test = script.

Test planning :-

In this phase, The automation test lead will be doing the foll :-

a) He will understand the requirements.

b) He will identify all the areas to be automated.

c) He will analyze both the positive and negative flow of the application.

d) He will prepare the tool ready with all the preconfigurational settings based on the dynamism of the application so that one can comfortably use the tool to test that application.

e) He will do The resource planning.

f) He will do the scheduling.

g) Finally he will prepare the automation

Generating the basic test :-

In this phase, the automation test engineers will be generating the basic test for both the positive and negative flow of the application.

Enhancing the test :-

↳ adding something to basic.

one can enhance the script in the foll- ways.

1. Inserting the check points
2. synchronizing the test
3. parameterising the test (Data driven testing)
4. Inserting the output values
5. Inserting the transaction points for measuring transactions.
6. Inserting the programatic statements.
7. Inserting the comments
8. writing the script statements manually.

Check points :-

Check point is a feature provided by the QTP which is used for checking something during the execution at any point of time.

Operational overview of checkpoint :-

Checkpoint works in 2 phases.

1. pre-execution phase
2. while execution phase.

1 Preexecution phase :-

In this phase, it will do the foll.

a) captures the expected value.

b) Generates the test script statement based on the expected value.

2. while execution phase :-

In this phase, it will do the foll.

a) It will capture the actual value.

b) It will compare the expected value with actual value.

c) Finally it will display the result.

Types of checkpoints :

1) standard checkpoint
2) Bitmap checkpoint
3) Text checkpoint
4) Text area checkpoint
5) Database checkpoint
6) XML checkpoint

} Visible checkpoints

Web checkpoints

7) page checkpoint
8) Table checkpoint
9) Image checkpoint

} Hidden checkpoints

10) Accessibility checkpoint } visible checkpoints

1) Standard checkpoint :

It is used for checking the standard GUI objects properties values. It can inserted in 2 ways.

a) Through application
b) Through active screen

Navigation through application :

Keep the cursor in desired location
↳ keep the tool under recording mode.
↳ activate the menu item 'Insert'
↳ Goto checkpoint
↳ select the option 'standard checkpoint'
↳ click on desired object in application
↳ click on 'ok'
↳ select the desired properties to be checked
↵
specify the desired expected values
↵
click on ok
↵
Stop recording.

Navigation through active screen :

Keep the cursor on the desired statement so that the corresponding snapshot will be available in the active screen

↳ Go to the active screen, right click on desired object

↳ select the option 'Insert standard check point'

↳ click on 'ok'

↳ select the desired propeties

↳ specify the desired expected values

↳ Select one of the foll. options

- before current step
- after current step

↳ click on 'ok'

Bitmap :

~~~ area formed with group of pixels.

Bitmap checkpoint :-

Bitmap checkpoint is used for checking the complete or a part of a bitmap. This checkpoint can be inserted in 2 ways.

1. Through application
2. Through active screen.
~~~

(Navigation through application):

Keep the cursor in desired location

↳ Keep the tool under rec. mode

↳ Activate the menu item 'Insert'

↳ goto checkpoint

↳ select the option 'Bitmap checkpoint'

↳ click on the desired bitmap in the application

↳ click on ok

if at all a part of a bitmap need to be selected then select it with the help of select area button.

If at all only that part need to be saved, select the checkbox

☐ save only selected area

click on 'ok'

Stop recording

* If fails then only it will display with diagrams in results

Navigation through Active screen:

Keep the cursor on the desired statement so that the corresponding snapshot is available in the active screen

↳ goto active screen rt. click on the desired bitmap.

↳ select the option 'Insert bitmap checkpoint'

↳ click on 'ok'

If all a part of bitmap need to be selected then select it with the help of select area button

↳ If at all only that part need to be saved, select the check box

☐ save only selected area.

↳

Select one of the foll. option

○ Before current step

○ after, current step

↳ click on ok.

8. Text checkpoint :

Text checkpoint is used for checking the text present on a specified object. It can be inserted in 2 ways.

1. Through application
2. Through active screen.

Navigation through application :

Keep the cursor in desired location

↓

Keep the tool under rec. mode

↓

Activate the menu item 'Insert'

↓

goto checkpoint → Text checkpoint

↓

click on desired object in application

↓

click ok

↓

Select ○ Before current step
○ After current step

↓

click ok

↓

Stop rec.

Keep the cursor in desired location

↓

Keep tool in rec. mode

↓ click on reqd. application

~~Activate menu item 'Insert'~~

↓ stop rec.

~~go to checkpoint — Text checkpoint~~

↓ go to Active screen

~~click on~~

↓

rt. click on the reqd text of the application

↓

click insert Text checkpoint

↓

click ok

↓

Select reqd. options like ☐ match case
☐ Ignore spaces
☐ Exact match
☐ Text not displayed

~~etc~~ select from

Insert statement ○ Before current step
○ After current step

↓

click on ok

↓

stop recording.

Text area checkpoint:

It is used to checking the text present in a specified area of the application

It can inserted only through the application

Navigation through application:

Keep the cursor in desired location

Activate ↓

Insert on menu bar

↓

go to checkpoint → Text area checkpoint

Select desired area with crosswires

↓

click OK

↓

Select or check the field options

- ☐ match case
- ☐ Exact match
- ☐ Ignore spaces
- ☐ Textnot displayed

↓

click OK

↓

stop recording.

Database checkpoint :-

Database checkpoint is used for checking the contents of a database.

Navigation :

Activate the menu item insert

↓

go to checkpoint, select 'database checkpoint' option

↓

Select the option ~~select the op~~ specify SQL statement manually,

↓

click on next

↓

click on create

↓

click on new

Select the desired driver

↓

click on next

↓

Specify a desired DSN (Ex: hyd26)
(Data source Name)

↓

click on next

↓

click on Finish

↓

click on select

↓

Browse the desired database

↓

click on OK

↓

click on OK

↓

click on OK

↓

Specify the desired SQL statement

↓

click on Finish

↓

Select the desired contents to be checked

↓

Select one of the foll. options

- O Before current step
- O After current step.

↓

Click on OK

XML checkpoint :-

↳ Extensible Markup Language.
universally understandable language
data transformation / transfer.

→ Extensible markup language.
→ It is a universally understandable language and used mainly for data transformation

XML checkpoint is used for checking the contents of an XML file.

Navigation :

Activate the menu item 'Insert'
↓
goto checkpoint
↓
select the option XML checkpoint
↓
Browse the desired XML file
↓
click on OK
↓
select the desired contents to be checked
↓
select one of the foll. steps
- O Before current step
- O after current step

↓
click on OK.

Page checkpoint:

Page checkpoint is used for checking the properties of a web page like Load time, No. of images and No. of links.

Navigation:

Keep the cursor on the desired statement so that the corresponding page is available in the Active screen (Google)

↓

goto the Active screen, rt. click on the desired page

↓

select the option "Insert standard checkpoint"

↓

Select the page in object heirarchy

↓

click on OK

↓

select the desired properties to be checked

↓

select one of the foll. options

- ○ Before current step
- ○ After current step

↓

click on 'OK'

Table checkpoint:

Table checkpoint is used for checking the contents of a table.

Navigation :

keep the cursor on desired statement so that the corresponding page is available in active screen

↓

goto Active screen

↓

rt. click on desired page

↓

Select the option insert standard check point.

↓

select table in object heirarchy

↓

click on ok

↓

select the desired prop. to be checked

↓

Select one of foll. options ○ Before
○ after

↓

click on ok

Image checkpoint :

Image checkpoint is used for checking the properties of an image.

Navigation :

Keep the cursor on desired statement so that the corresponding pg is available in active screen

↓

goto Active screen

↓

rt. click on desired page

↓

select option insert standard check pt.

↓

select image in obj heirarchy

↓

click ok

↓
click OK

Accessibility checkpoint:

Accessibility checkpoint is used for checking whether the webpage is developed according to the w3c standards or not inorder to

↓

world wide web consortium

confirm whether that page can be accessible comfortably all over the world or not.

Navigation:

Keep the cursor on the desired statement so that the corresponding page will be available in the Active screen.

↓

go to Active screen, rt. click on desired obj. select the option, Insert accessibility checkpoint

↓

Select one of the foll. options

- ○ Before current step
- ○ After current step

↓

click on OK.

Synchronization:-

synchronization is a process of matching the speeds of both the tool and the application inorder to keep them in sync with each other to obtain proper testing results.

Here the main concept is making the tool to wait till the application finishes its work. It can be done in 3 ways.

1. Inserting the synchronization point
2. Increasing the default time
3. Inserting the wait statement directly in the script.

Navigation for inserting synchronization pt.

Keep the cursor in the desired location
↓
Keep the tool under rec. mode
↓
Activate the menu item 'Insert'
↓
Select the option 'synchronization pt'
↓
Specify the desired object in appl.
↓
click on ok
↓
Specify the desired prop. name & value
↓
Specify the desired extra time
↓
click on ok
↓
Stop recording.

To avoid the above navigation, one can directly type the foll. script statement :—

* objectheirarcy.wait properly ("prop.name", prop.value, extra time in millisec)

Navigation for increasing default time

Activate menu item 'File'

↓

Select the option 'settings'

↓

Select Run tab

↓

Specify the desired time in the obj synchronizatio timeout field.

↓

click on 'apply' and 'ok'

Syntax for wait statement

wait (time in sec)

—

Data Driven testing :

↳ Retesting

Data driven testing is a concept introduced in automation inorder to implement retesting.

steps to be perform data driven testing :

1. Collect the data into the data table.
2. generate the basic script.
3. Do the required enhancements and parametrise the script.
4. Execute the script.
5. Analyse the results

parametrization :

parameterization is a process of replacing the constant values with variables in order to increase the scope of the test.

↳ make it powerful

parameterization can be done in 3 ways.

a) Through data driver wizard
b) Through keyword view
c) Manually.

Navigation thru datadriver wizard

Activate the menu item 'Tools'

↓

select the option 'data driver'

↓

Select the desired constant value

Diff. Data sources like excel, db... why datatable no

File without having any specific format
— flat files.
ex: notepad.

Variable :

* Whenever u r entering data in notepad, give tab space then QTP will understand that it is in next column.
* "Const" variable

---

click on parameterize
↓
click on next
↓
click on parameter options button
↓
specify desired col. name "v1"
↓
click on ok
↓
click on Finish

In the same way parameterize all the constant values reqd. and finally click on ok.

Navigation Thru keyword view:

Go to keyword view
↓
Select the desired const. value
↓
click on configure the value button
↓
select the option 'parameter'
↓
select the desired col. name
↓
click on ok.

To avoid the above navigation, one can directly type the script manually as follows :-

```
x = datatable ("v1", 1)
vbwindow ("calculator"). vbEdit ("value 1"). set x
vbwindow ("calculator"). vbEdit ("value 2").
                          Set Datatable ("v2", 1)
vbwindow ("calculator"). vbbutton ("Add"). click
```

Navigation for parameterizing the checkpoint

Rt. click on checkpoint statement
↓
or select the option 'checkpoint properties'
↓
Select the option 'parameter'
↓
click the parameter options button
↓
Specify the desired column name

Click on OK
↓
Click on OK.

Datatable :

Datatable is a important feature provided by the QTP which provide the foll. facilities:

a) It is used for holding the test data

b) It isolates the test script from the data source

c) It provides the provision to import the data from any datasource like XL files, databases or flat files very easily.

d) It allows the user to interact with the data directly.

Note : QTP maintains 2 copies of datatables.
1. Design time data table
2. Runtime datatable

Output value :-

Output value is a feature provided by QTP which is used for capturing a value from an application or from a database or from an XML file and store it under a specified column in the runtime data table during execution.

Types of output values:

There are 5 types of output values. It is used capturing standard GUI object's property's values.
1. standard output value

2. Text output value

It is used for captured the text present on a specified object

3. Text area output value

It is used for capturing the text present in a specified area.

4. Database oulput value.

It is used for capturing a value from the database.

5. XML oulput value.

It is used for capturing a value from XML file.

Navigation for standard o/p value :

Keep the Tool under rec. mode

↓

Activate the menu item Insert

↓

Goto oulput value

↓

select the option 'standard o/p value'

↓

click on desired object in application

↓

click on ok

↓

Select the desired prop.

↓

Modify col.name if reqd

↓

Navigation through active screen

Keep the cursor on desired statement so that the corresponding screen is displayed in AS.

↓

Goto AS

↓

or click on desired obj.

↓

Select option Insert o/p value

↓

Click on ok

↓

Select the desired property

↓

Modify the name if reqd.

↓

Select one of the foll options

- before ~~aft~~ current step
- after current step

↓

Click on ok

Measuring Transactions

By Inserting Transaction points

Measuring Transaction is a concept provided by QTP which is used for calculating the time taken by an application to perform a specific task or the execution time of a block of statements.

To do the same, QTP has provided 2 options.

1. start transaction
2. End transaction.

Navigation

Activate the menu item 'Insert'

↓

Select the option start transaction / end transaction

↓

Specify transaction name

↓

Select one of foll. option

- ○ Before current step
- ○ after current step

↓

click on ok

To avoid the above navigation, one can directly type the foll. script.

```
services. starttransaction "tr1"
services. Endtransaction "tr1"
```

("tr1" → Transaction name; services → script; starttransaction / Endtransaction → utility statement.)

Inserting the programatic statements

(programatic → The statements present in a program.)

Vb script program contains the foll. programatic statements.

1. Normal statements / object calls
2. Conditional statements
3. looping statements
4. Utility statements

Note: Apart from these types of statements, usually ... available

Utility object

System util object is mainly used for invoking the application with the help of run method.

syntax

systemutil.run "path of the application"

Note: One can invoke the application using the invoke application statement also.

syntax

invokeapplication "path of the application"

Reporter utility object

Reporter utility object is used for sending the user defined steps to the result window with the help of report event method.

syntax:

Reporter.Reportevent status, "reportname", "Details"

Ex:

Reporter.Reportevent 2, "myrep", add operation is successful.

Status

0 — passed
1 — Failed
2 — Done
3 — warning

Debugging phase :

Debugging is a process of executing the script or a program in a user desired fashion with some temporary breaks inorder to identify the errors.

To do the same, QTP has provided the step commands and break point feature.

step commands :

1) step into : It is used for executing a single step if that step is a function call step then it will make the pointer enter into the function and stops the execution.

2) step out : once the pointer is inside the function, from the position of the pointer if at all, all the remaining statements need to be executed in the function and stop the execution by stepping out of the function then we will use step out command.

3) step over : step over is used for ~~execu~~ until any step is over (finished) i.e., it is used for executing a single step. If that step is a function call step, then it will execute all the statements inside the fn and then stops the execution.

Breakpoint is a feature provided by QTP which is used for breaking the execution temporarily.

Run to step:

It is used for running all the steps upto the position of the cursor.

Debug from step:

It is used for moving the pointer to the statement where the cursor is available by skipping all the previous statements.

Debug viewer pane: → area

[Area provided by QTP for viewing, modifying, setting the current values of variables during execution break.]

Debug viewer pane is an area provided by QTP which is used for viewing, modifying or setting the current values of the variables or expressions during the execution break with the help of 3 tabs by name watch, variables and command.

Executing the test:

In this phase one will execute the test.

Analysing the results

In this phase one will analyse the results

Recovery scenarios :

↳ getting back to original profile.

- qrs
  ↓
  quick recovery scenario.

QTP – execution – prob. occured – still it performs – then warning

notepad •vbs (vbscript) → lib. file → To call fn.
(or)
•qfl quick fn library

} fns need to be stored in library files.

* prob. with application – wait for 20s
  use sol. after 20s.

with script – doesn't wait

During the execution, sometimes QTP will face problematic situations wherein it need to recover by itself and continue the execution. To do the same, we need to prepare the solution well in advance and make it available to that test before the execution starts.

QTP may face the foll. problematic situations.

1. problems related to pop up window
2. problems related to object state
3. problems related to testscript execution
4. problems related to Application crash.

To define the solutions for above problems,

Trigger event :-

1. Popup window
2. object state
3. Test run error
4. Application crash

Navigation for defining The solution for object state problem

Activate The menu item resources

↓

Select "Recovery scenario manager" option

↓

click on new scenario

↓

click on next

↓

Select The option obj. state

↓

click on next

↓

specify the desired obj. with the help of hand icon.

↓

click on ok

↓

click on next

↓

specify the desired properties

↓

click on ok

↓

click on next

↓

click on next

↓

Select the option fn call

↓

click on next

↓

Browse the desired library file

↓

specify the desired fn name

↓

write the reqd. script in the body of the fn

↓

click on next

↓

deselect
~~Remove~~ the check box

☐ Add another recovery operation

↓

click on next

↓

Select one of the req. post recovery options

↓

click on next

↓

Specify the desired scenario name

↓

click on next

↓

Select the reqd. checkboxes

☐ Add scenario to current test

☐ " " " default " settings

↓

click on finish

↓

click on save

to the current test

Activate the menu item File
↓
Select the option 'settings'
↓
Select the recovery tab.
↓
Browse the desired recovery file
↓
click on Add scenario
↓
click on apply and ok.

Batch Testing / Batch Execution

- mtb – mercury test batch (previously)
  - now it is module because HP took over.

Executing a group of tests continuously at a time is known as batch execution or batch testing.

Usually this concept will be very much useful during regression testing. To do the batch testing, QTP has provided a special tool by name test batch runner.

using the test batch runner we can create the batch files & execute them.

*** Note: Batch file extension is .mtb

Navigation
↓
next page

Nagivation for test batch runner

start → programs → QTP → Tools → Test batch runner.

Environment variables

↓ ? why ?

usually we prefer EV whenever

holds the values change from time to time

Adv :- AT is like investment we will get returns in future.

1) Whenever we are developing the automation test scripts, we know that these scripts are not going to be just used in the testing environment but also in different environments in future.

2) So while developing the test scripts, whenever we feel that some common variables need to be used in many tests, then we ~~will be~~ should declare those variables as environment variables in an environment file seperately, attach that environment file to the reqd. tests and should use the variables inside the tests. 3) By doing this if at all any of the variables values need to be updated in future, whenever the environment is changed, one can very easily & fastly update the values in the environment file &

There are 2 types of environment variables.

1. Built-in variables
2. user defined variables

Built-in variables by default will be available with every test.
So one can directly use them in any test in the foll. way.

syntax :

environment.value("Built in variable name")

Ex:

```
Var = environment . value ( " productver")
msgbox var
```

output - 9.2

Whenever we feel that we need to declare some variables based on our application, then one can define the user defined variables.

The user defined variables file need to be attached (associated) to the reqd. test & then only we should use them inside the tests.

Navigation for creating an environment file with user defined variables

Activate the menu item 'File'

↓

Select the option 'settings'

↓

Select the environment tab

↓

Select the variable type as user defined

↓

O

Add all the reqd. variables and corresponding values

↓

click on export button

↓

Browse the desired location as save the file with .XML extension.

↓

click on Apply & OK.

To avoid the above navigation, one can directly prepare xml file as follows.

open Notepad

↓

develop the foll. script

↓

Save it with .xml extension

```
< Environment >
    < variable >
        < name > a </ name >
        < value > 10 </ value >
    </ variable >
    < variable >
        < name > apppath </ name >
        < value > "c: \cal.exe" </ value >
    </ variable >
    < variable >
        < name > b </ name >
        < value > 20 </ value >
    </ variable >
```

Navigation to associating an environment file to the reqd test

Activate the menu item 'File'

↓

Select the option 'settings'

↓

Select Environment tab

↓

Select variable type as user-defined

↓

Select the check box Load variables & value from external file

↓

Browse the desired environment file

↓

click on apply & OK.

To avoid the above navigation, one can directly type the foll. statement in the script.

Syntax

environment.loadfromFile ("path of the environment file")

Ex:

```
environment.LoadFromFile "c:\...\*.xml"path"
x = environment.value ("a")
Y = environment.value ("b")
P = environment.value ("apppath")
systemutil.Run P
Vbwindow ("calculator").VbEdit("val 1").Set x
Vbwindow ("calculator").VbEdit("val 2").Set Y
Vbwindow ("calculator").Vbbutton ("SUB").click
```

Regular Expressions

To avoid popup windows.

. Anything

* All

Whenever the GTP is unable to identify the objects during the execution due to regular changes in the object's property's values then to overcome this problem, one need to replace the corresponding const. value in the obj. repository with a suitable regular expression.

Navigation

Open obj. repository

↓

Select the desired obj. in obj. heirarchy

↓

Select the desired prop. in prop.'s list

↓

click on configure the value button

↓

Replace the const. value with suitable regular expression

.*

↓

select the checkbox regular expression

↓

click on ok

parameterization should be done

Fax order

after sending no changing

To recognize this is used.

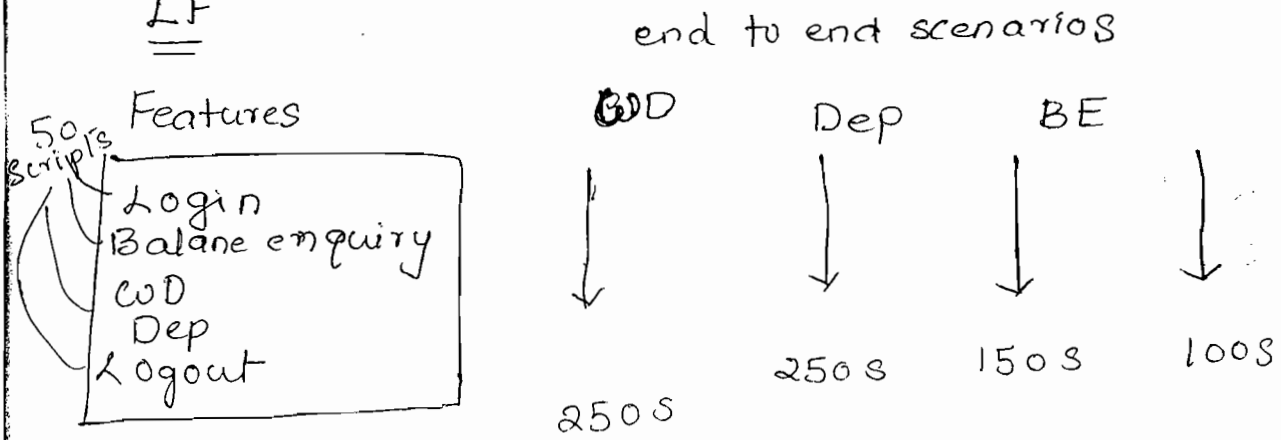
Frameworks :-

Framework is a generic work or set of guidelines designed by group of experts and followed by many people to perform a task in an effective, efficient and optimised way.

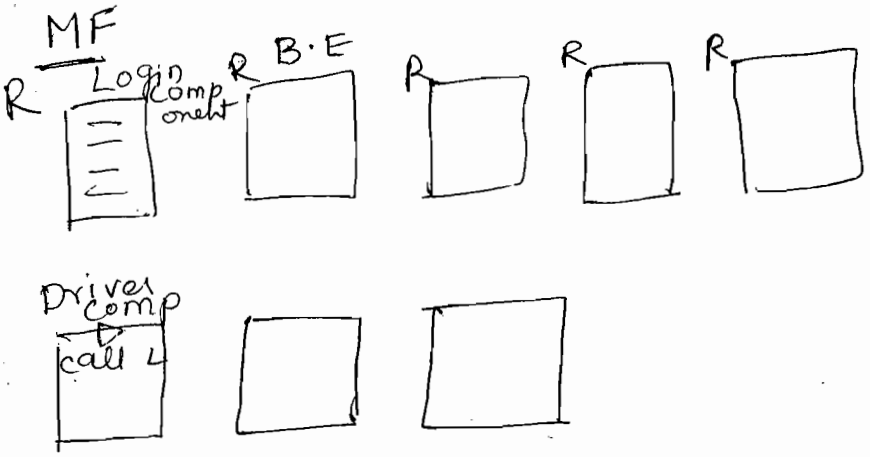
Few
* Types of Frameworks :

1. Linear Framework
2. Modular Framework
3. Keyword driven Framework
4. Hybrid Framework
5. Data driven Framework

LF



Redundancy is more



Ex: on Flight Application

| Action — 2 actions at a time.

** GTP 9.5

↓

all actions are reusable

#w call to copy of action

diff

Linear Framework :-

1. Generate the basic test.
2. Enhance the test
3. Debug the test
4. Execute the test
5. Analyze the results.

Modular Framework :

1. Develop the individual components for the tasks that are to be performed.
2. Make them as reusable components
3. prepare the driverscripts for end-to-end scenarios.
4. Execute the drivers
5. Analyse the results.

Flight appl.

↓

enter name : shanti
pwd : mercury

↓

click Login

↓

Fill all the data in FR window

↓

Insert order

↓

File → open order

↓

enter ord. NO – 6

↓

close window

↓

stop recording.

Now seperate each action with split action.

↓

Now make each action as reusable action

↓

save the file.

Now open New Test

↓

goto Insert – call of existing action

↓

make driver script

↓

run

↓

analyse results.

Actions :-

Action is defined as a set of script statements to perform a specific task.

Types of Actions :-

2 Types

1) Normal,
2) Reusable actions

Note :* The reusable actions called in another tests are known as external actions.

* External actions are non-editable.

Operations on actions

1. Inserting a new action

Activate menu item 'Insert'
↓
select The option 'call to new action'
↓
specify desired action name
↓
select one of The foll. options
1. at end of test
2. after current step
↓
click on OK

2. Splitting an action into 2 actions.

Save The action
↓
Keep the cursor on The 1st line of the second part
↓
activate menu item 'edit'
↓
..

Select the option 'split action'

↓

Select one of the foll. options

- O Independant of each other
- O nested.

↓

Specify the desired names for the actions

↓

click on ok.

3. Making an action as a reusable action.

Select the desired action

↓

activate the menu item edit

↓

goto action

↓

select the option 'action properties'

↓

select the check box Reusable action

↓

click on ok

4. Renaming an action

Select the desired action

↓

activate the menu item 'edit'

↓

goto action

↓

Select the option 'Rename action'

↓

specify the desired name

↓

click on ok

5. calling an existing action

Activate the menu item 'Insert'

↓

select the option 'call to existing action'

↓

Browse the desired test

↓

Select the desired action

↓

select one of the foll. options

- O at the end of the test
- O after the current step

↓

click on ok

***

V. I

make data files in XL files & save it

Shared repositories done in 2 ways.

Resources
↳ obj. rep. manager
↳ add

(or)

Resources
↓
obj. repository
↓
local objects

TL
↓
create
~~software~~ folder (parent) → child → child
structure

- easy to access

***** V. Imp

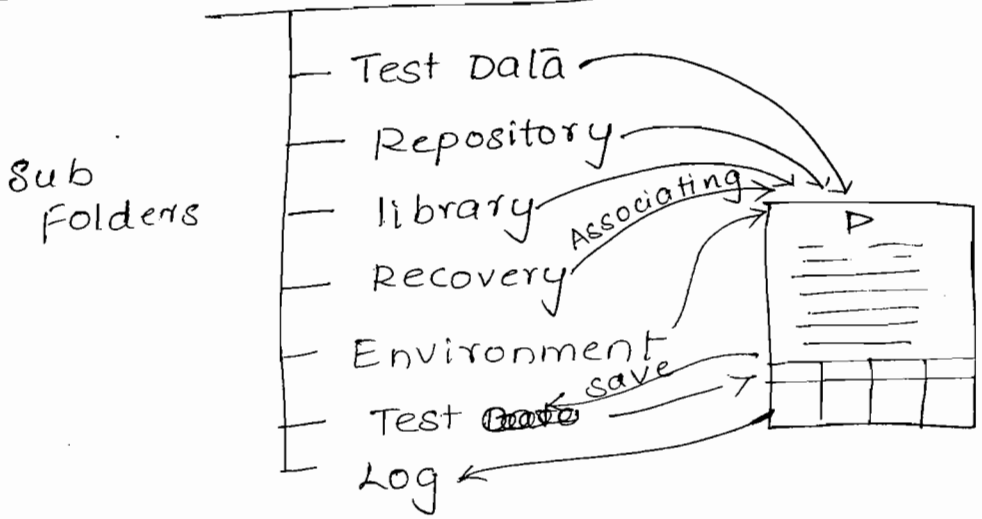
Automation Testing process with Keyword Driven Framework

1. TL (Test Lead)

↓

creates Folder structure

Filename :- PN (proj. name) – Automation



2. create reqd. Test data files and save them in corresponding TD folder.

3. create reqd. shared repositories & save them in corresponding folder.

4. create reqd. library files & save them in the corresponding folder.

5. Create the reqd. recovery files & save them in corresponding folder.

6. Create the reqd. environment files & save them in corresponding folder.

7. open the main test.

8. Associate all the reqd. resources.

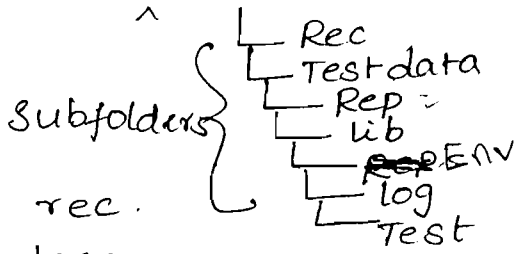
9. Develop the script in such a way that it will be executed based on the keywords specified in the datatable.

10. Save the test in the corresponding folder open it whenever reqd. specify the desired keys and execute it.

11. Finally analyze the result.

Ex script

```
Var = datatable ("keys", 1)
select case var
case "l1"
   call login()
     insord ()
     logout ()
case "l2"
    call login ()
    opnord ()
     logout ()
case "l3"
    login ()
    logout ()
End select.
```



↓ start rec.

open flight window

1) enter login name
2) pwd.

enters FR window

Insert reqd. data – insert order

File – openorder then close application.

Stop recording

↓

Now since Local rep. gets created, we can export to shared repository.

(or)

If we didnot record above application, we can just create shared rep. as follows.

resources

→ obj. rep. manager

→ Click add button

→ show icon on login window & FR window & open order

store them one by one & save it as shared rep. with .tsr.

↓

Now make ~~library~~ fns in library file and store it with .vbs extension.

↓

Now make env. file & save with .xml extension

File

↓

settings

↓

environment

↓

add ⊞ apppath

path of login window

Now save this in Test folder.

Now Take New file
start calling ~~tos~~ as follows -

Ex script

↓

Datatabe

Keys
l_1
l_2
l_3

↓

Now associate shared rep. as follows.

Resources
↳ associate rep
↳ [+]
↳ click on Action1
↳ click [>]
↳ click ok

↓

Now associate lib. files as follows.

File
↳ settings
↳ Resources
↳ [+]
↳ click on desired file
↓
open
↳ apply
↳ ok

↓

Now associate environment to file

File
↳ settings
↳ environment
↳ ☑ Load variables and values from external file
Browse the lib. file.
↳ apply
↳ OK.

↓

Add
system.util.run "path of application"
(or)
env. variable

inorder to invoke by itself.

↓

Run

↓

Analyse the results.

file

Writing fns

Function Login()

- - - -

- - - -

End Function

Function Insord()

- - -

- - -

End Function

Function opnord()

- - - -

End Function

Function Logout()

End Function.

Regular expressions

invoke Login

↓

start rec.

Enter details, FR window

↓

open order, ord NO - 6 eg

↓

Fax order

||| ||| ||||

↓

send

↓

close

↓

Stop rec.

↓

Now parametrize

change set "6" to set Datatable ("ord_NO", 1)

Fill ord. NO.'s

↓

Run.

Hybrid Framework

Mixture of any 2 or more frameworks is known as Hybrid framework.

Library files

Library file is an area where one can develop & store the user defined fns

Navigation for preparing a lib. file.

open notepad
↓
write all the reqd. fns
↓
save it with .vbs extension

II Navigation.

Activate the menu item 'New'
↓
Select the option fn library
↓
develop all the reqd. fns
↓
save it with .qfl or .vbs extension.

Navigation for Associating library file to the reqd. test.

Activate the menu item 'File'
↓
select the option 'settings'
↓
Select resources Tab
↓
Add the reqd. library file
↓
click on apply & OK.

keys for handling the whole project maintained in 3 sheets

by Testlead.

Sheet 1

MID	STATUS
M_1	Y
M_2	N
M_3	N

sheet 2

TSID	MID	STATUS
TS_1	M_1	N
TS_2	M_1	Y
TS_3	M_1	N
.	.	
.	.	
TS_{10}	M_1	N
TS_{11}	M_2	N
TS_{12}	M_2	N
.	.	
.	.	
TS_{18}	M_2	N
TS_{19}	M_3	N
.	.	
.	.	
TS_{25}	M_3	N

Sheet 2

TcID	TSID	Keys
Tc1	TS1	Login
TC2	TS1	Insord
TC3	TS1	Logout
TC4	TS2	Login
TC5	TS2	opnord
TC6	TS2	Logout
TC7	TS3	Login
TC8	TS3	Logout

Virtual object Configuration

Recognization →

Identification →

Virtual object configuration is a process of making the QTP to treat a selected area in the application or an user defined object as a specific object.

usually we need to do this configuration whenever the QTP is not recognizing the objects in your application.

* unless we delete from the VOC, QTP will remember.

Navigation for virtual object configuration

Activate the menu item 'Tools'

↓

Goto virtual objects

↓

Select the option 'new virtual object'

↓

click on Next

↓

Select the desired class of the object

↓

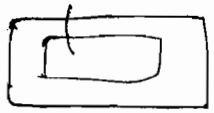
click on next

↓

Select the area in your application with the help of mark obj. button

↓

Ex



Select one of the foll. options.

- o Entire parent heirarchy
- o parent only.

↓

Click on next

↓

Specify the desired obj. name

↓

Specify the desired collection name

↓

click on Finish.

Note: Virtual obj. manager is a feature provided by GTP which is used for creating and managing the virtual objects.

Once created virtual obj. will be permanently remembered by the GTP unless until we delete it.

Vb Script

Variable

Variable is a container or a name given to a memory location which can hold a value and that can be changed in future whenever it is reqd.

Advantages of variable :

1. Reusability
2. Easy to maintain

Declaration of Variables :

one can declare the variables in the Vbscript with the foll. syntax.

dimension Dim variablename
private variablename
public variablename
} Vbscript

Note: public variables will be usually declared in the external files and can be used in any no. of tests by attaching that external file to those tests.

env variables — provided by QTP

Naming convensions of variable

1. variable name should be meaningful.
2. variable name should start with an alphabet.
3. It should not exceed 255 characters.
4. It should not include period (.)
5. It should not be same as reserved words or Keywords

for ↓

Option Explicit statement

Option Explicit statement will restrict the usage of new variables in the script without declaring them.

So good programers will always start their program with option explicit statement, declare the required variables & then use them in the script.

Ex

```
OPTION EXPLICIT
Dim APPPATH
DIM - ...
DIM - ...
APPPATH = "D:\cal.exe"
- - -
. . .
Invoke Application APPPATH
=========
=========
=========
Invoke Application APPATH
=========
=========
=========
Invoke Application APPPATH
```

The benefit is any misspelled variable or new variable is found during the execution it will show us an error

APPATH → misspelled

if misspelled QTP will show error if option explicit is declared otherwise it will not show.

Array

multi — 60 dimension
subscripts.

Array is a special type of variable which is used for holding many values.
There are 2 types of arrays.

1. single dimensional arrays
2. Multi " "

we can declare the arrays as fixed size arrays or dynamic arrays.

- These arrays can hold the fixed no. of values.

eg: DIM a(10)
a(0) = suresh
a(1) = Ramesh
:
:
a(9) = . . .

- This array can hold 10 values only.

- These arrays can be resized whenever it is reqd. in the program.

eg: DIM a()
=====
REDIM a(20)
=====
REDIM PRESERVE a(35)

- First we need to declare the dynamic array.

DIM var.name()

whenever we want to use this variable, we need to give the size of array with help of redim statement as follows.

REDIM var.name(size of the array)

then starts storing.

whenever we want to resize the array again we can use the same REDIM statement but if u want to preserve the previous location and extend the locations then we need to use PRESERVE keyword along with REDIM statement as foll:

REDIM PRESERVE var.name (size)

Multidimension arrays can be declared as follows :-

Syntax: Dim var.name (subscript 1, subscript 2, ... subscript60)

Ex: Dim a(10,10)

a(0,0) = ___

a(0,1) = ___

a(0,2) = ___

till Hundred values

Note: we can declare the multidimensional arrays upto 60 subscripts in Vbscript.

Operators :

Arithmetic operators

+, −, *, /, mod, ^

comparision operators

<, >, >=, <=, <> not equal to

concatenation

&

There are 5 types of operators.

1. Arithmetic operators

+, −, /, *, mod, ^ → exponentiation

10 mod 3 = 1
15 mod 6 = 3
10 ^ 4 = 10^4 = 10,000

2. Assignment operator

=

3. Concatanation operator

&

two strings

4. Comparison operators

<, >, <=, >=, =, <>

5. Logical operators

AND

OR

NOT

EXOR

AND

expression

E_1	E_2	Result
T	T	T
T	F	F
F	T	F
F	F	F

OR

E_1	E_2	Result
T	T	T
T	F	T
F	T	T
F	F	F

XOR

E_1	E_2	Result
T	T	F
T	F	T
F	T	T
F	F	F

NOT

E_1	Result
T	F
F	T

Conditional statements

There are 2 types of conditional statements.

1. If

~~then~~ then

else

2. Select case

If Then Else

This conditional statement is used for executing either the first block of statements or second block of statements based on the condition. ~~of statement~~

syntax

```
If (condition) Then
    ===
    ===
    ===
else
    ===
    ===
    ===
End If
```

Select case statement

This statement is used for ~~selecting~~ executing a particular block of statement among many blocks.

syntax

```
Select case exp
    case 1
    ===
    ===
    case 2
    ===
    case 3
    ===
    case else
```

Looping statements

Looping statements are used for executing a block of statements for certain number of times repeatedly continuously.

1) For ..... Next loop

Ex:
```
For i = 1 to 10 steps
   ---
   ---
   ---
Next
```

If is used for executing a block of statements for fixed no. of times.

```
For var = 1 to number step no.
   . . . .
   . . . .
   block of statements
   . . . .
   . . . .
Next
```

step NO — Increment

depending on step NO., it will increment by that. Default is 1.

1 to 10 step 3

means 1, 4, 7, 10

2) while ..... wend

It is used for executing a block of statements repeatedly till the condition is being satisfied & whenever the condition is not satisfied, it will come out of loop.

```
while (condition)
   . . .
   . . .
   ---
wend
```

3) dowhile ... loop

This loop works same as whilewend ... loop. The people who are convinient with this syntax will use this otherwise they can use the whilewend ... loop.

```
do while (condition)
  - - -
  - - -
Loop.
```

4) do.... loopwhile

This loop will work same as above loop but first time execution will be done without checking the condition.

```
do
  - - -
  - - -
Loop while (condition)
```

5) dountil..... loop

This loop is used for executing the block of statements repeatedly continuously till the condition is not being satisfied. once the condition is satisfied it will come out of the loop.

Syntax

```
do until (condition)
  ===
  ===
Loop.
```

c) do ..... loop until

This loop works same as above loop but the first time execution will be done without checking the condition.

syntax

```
do
  ---
  ---
Loop until (condition)
```

Procedures

There are 2 types of procedures.

1. subprocedures — execution fast
2. Function procedures

Whenever a task need to be performed again and again in different situations, then we will define a sub procedure or a fn procedure with the block of script statements which can perform that task and call the reqd. procedure in the main test wherever reqd.

Note: ① usually, the procedures will be defined in the library files and that library file will be attached to the reqd. test before using them.

② The Function procedure will work same as sub procedure but it can also return a value which the sub-procedure cannot give

③ sub procedure execution speed is more than the fn procedure's execution speed. so whenever there is noneed to return a value, we'll prefer subproce

But whenever we need to return a value, compulsarily we need to use the fn procedure.

(4) The value that is to be returned from the fn need to be stored in the fn name itself.

<u>syntax for sub procedures</u>

```
Sub procedurename (arguments)
    ___ ___
    _ _ _
    _ _ _ _
End sub
```

Ex

```
sub Login (UN, Pwd)
    - - - - -  set UN      }
    - - - - .  set Pwd     }  library file.
    - - - - (login) .click }
End sub.
```

Main Test

```
_______________
_______________
_______________
Login ("suresh", "mercury")
_______________
_______________
_______________
Login ("chiru", "sridevi")
```

<u>order</u>

1. Arithmetic
2. Assignment

Syntax for fn procedure

```
Function procedurename(arguments)
  ---  -----
  ---  ----  ---
  ---  ----  ---
  -----        ---
End Function
```

Ex:

```
Function Add (a,b)
    c = a+b
    Add = c
End Function
```

returns 'c' ie a+b

Main Test

```
----------
----------
var = Add (10,20)
----------
var1 = Add (30,80)
----------
var2 = Add (60,40)
```

The Regularly used methods with some real time scenarios

1. capture Bitmap

capture Bitmap is a method which is used for capturing the snapshot of any obj. & storing it in the desired location.

Syntax

Object heirarchy. capture Bitmap "path of the location with a filename.bmp" extension

Ex:

```
window ("FR"). Dialog ("open order"). winButton ("ok").
  captureBitmap "d:\fl_auto\def2.bmp"
```

2. Exist method

Exist method is used for checking whether the object is existing or not.

If at all the obj. is existing, then it will return 'True'. If at all the object is not existing, then it will make the tool to wait till the obj. exist and whenever the obj. exists then it will return True & allow the QTP to continue the execution.

In case if the obj. doesnot exist upto the max. time (default time + extra time) then finally it will return false.

Syntax:

Var = objectheirarchy.exist [(Time in sec)] ← means optional.

Ex:

Var = Vbwindow("cal").Vbbutton("Add").exist

or

" .exist(15)

3. wait properly

wait prop. method is used ~~for~~ to wait based on the object's property's value or upto the max. time.

Syntax

objectheirarchy.wait property ("prop. name", prop. value, Extra time in millisec)

Syntax: wait (Time in sec)

It is used for making the tool to wait till the specified time is elapsed.

4. Click method

Click method is used for clicking on a specified object.

Syntax: obj.heirarchy.click [x,y,[button]]

0 – left click
1 – rt click
2 – middle click.

5. DBL click

Double click is used for doubleclicking on an object.

Syntax: obj.heirarchy.dblclick [x,y,[button]]

0 – lt
1 – rt
2 – middle

6. Set method

Set method is used for performing operations usually on the foll. objects.

1. Edit box
2. check box
3. Radio button

Set method with edit box

It is used for setting any value into an edit box

syntax

obj.heirarchy of editbox . set "value"

set method with check box

It is used for selecting or deselecting the check box.

syntax

obj.heirarchy of checkbox . set "ON" / "OFF"

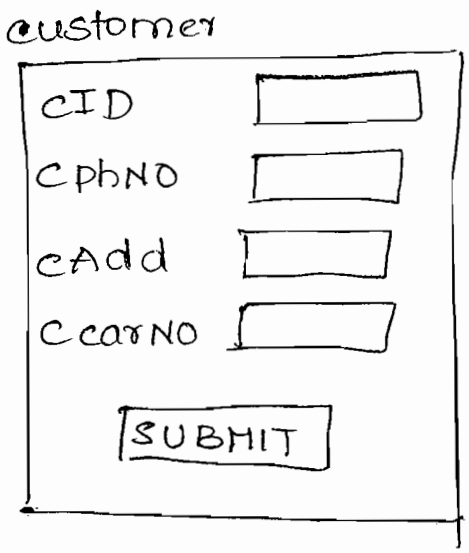
Set method with radio button

It is used for selecting a radio button among a group of radio buttons.

syntax

obj.heirarchy of radiobutton . Set

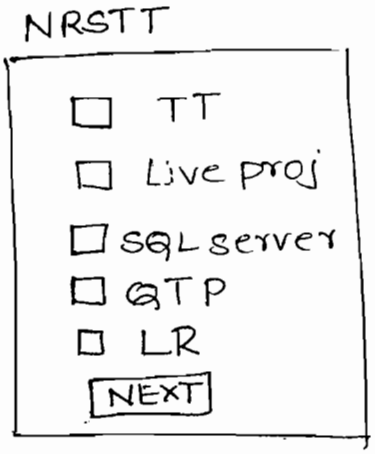
Case study - 1



007
11111
HYD
1111

```
Vbwindow("customer").VbEdit("CID").set "007"
Vbwindow("customer").VbEdit("cphNO").set "11111111"
Vbwindow("customer").VbEdit("CAdd").set "Hyd"
Vbwindow("customer").VbEdit("ccarNO").set "1111"
Vbwindow("customer").Vbbutton("SUBMIT").click
```

Case study - 2

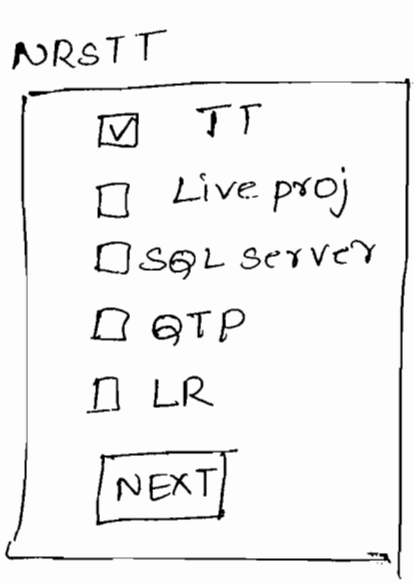


Develop the script for seelecting TT check box, Lproj. checkbox, SQLserver checkbox & clicking on Next button.

```
Vbwindow("NRSTT").Vbcheckbox("TT").set "ON"
Vbwindow("NRSTT").Vbcheckbox("Liveproj").set "ON"
Vbwindow("NRSTT").Vbcheckbox("SQLserver").set "ON"
Vbwindow("NRSTT").VbButton("NEXT").click.
```

* No need to off already deselected checkbox

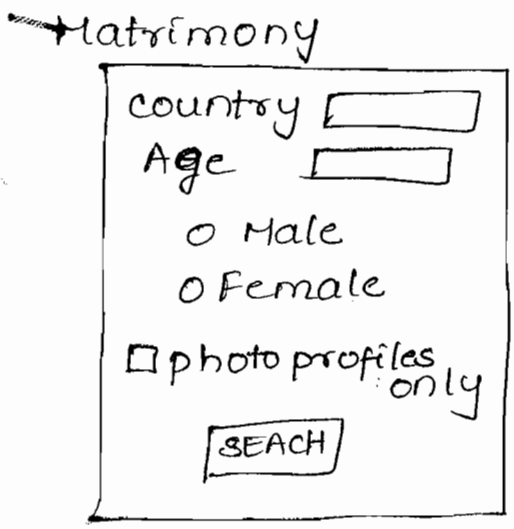
Case study 3



Develop the script for deselecting the TT checkbox, selecting the liveproj, QTP & LR checkboxes & clicking on Next button.

```
Vbwindow("NRSTT").Vbcheckbox("TT").set "OFF"
Vbwindow("NRSTT").Vbcheckbox("Liveproj.").set "ON"
Vbwindow("NRSTT").Vbcheckbox("QTP").set "ON"
Vbwindow("NRSTT").Vbcheckbox("LR").set "ON"
Vbwindow("NRSTT").VbButton("NEXT").click
```

Case study 4



Develop the script for entering country, age, selecting the gender, selecting the checkbox photo profiles only & clicking on search button.

```
Vbwindow("Matrimony").vbedit("country").
                                set "India"

Vbwindow("Matrimony").vbedit("age").set "28"
Vbwindow("Matrimony").vbradiobutton("Male")
```

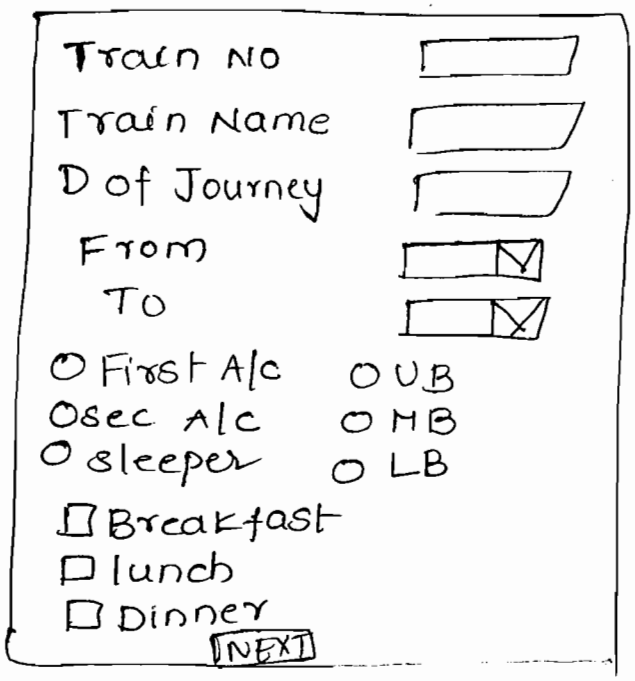
Vbwindow("matrimony").Vbcheckbox("photoprofiles only").set "ON"

Vbwindow("matrimony").VbButton("search").click.

Case study 5

Develop the script for entering TNO, Tname, D of J, selecting the starting & destination points selecting the reqd class, selecting the reqd berths selecting the BF & dinner check boxes, and clicking on Next button.

Res




```
Vbwindow("Res").VbEdit("Train NO").set "1234"
Vbwindow("Res").VbEdit("Train name").set "Palakrna"
Vbwindow("Res").VbEdit("DOJ").set "19/2/09"
Vbwindow("Res").Vbcombobox("From").select "Hyd"
Vbwindow("Res").Vbcombobox("TO").select "Vizag"
Vbwindow("Res").Vbradiobutton("First A/c").set
Vbwindow("Res").Vbradiobutton("LB").set
Vbwindow("Res").Vbcheckbox("BF").set "ON"
Vbwindow("Res").Vbcheckbox("Dinner").set "ON"
Vbwindow("Res").VbButton("Next").click.
```

Select Method

Select method is used for selecting an item from a combo box or list box.

Syntax

obj.heirarchy.select "Item"

Activate Method

Activate method is used for activating a dialog or a window.

Syntax

obj.heirarchy.activate

close Method

close method is used for closing a coindoco or a browser.

Syntax

obj.heirarchy.close.

Set secure Method

Set secure method is used for setting The encrypted data into an edit box.

Syntax:

obj.heirarchy.setsecure "encrypted string"

Encrypted string can be generated with the help of a special tool by name password encoder.

Navigation for password encoder:

start → programs → QTP → Tools → password encoder.

Type method

mic - microsoft integer conversion

Type method is used for making the QTP to perform the keyboard related operations.

syntax

Obj.hierarchy.Type Keyvalue

EX

Dialog("Login").WinEdit("AgentName:").
Type micTab

Syntax check

Syntax check is a feature provided by QTP which is used for checking the syntax errors in the script. Tools → syntax check

Information pane

Information pane is used for viewing the syntax related information after the syntax check.

Missing Resources pane

This is from QTP 9.0 onwards

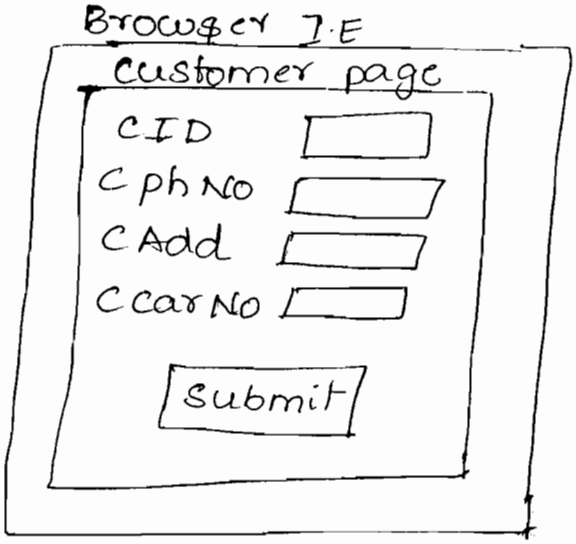
While opening a test, if at all any associated resources are missing then that information will be shown in the missing resources pane.

↓

Test is saved after 6 months again we want to open, by that time somebody deleted associated info/. resources

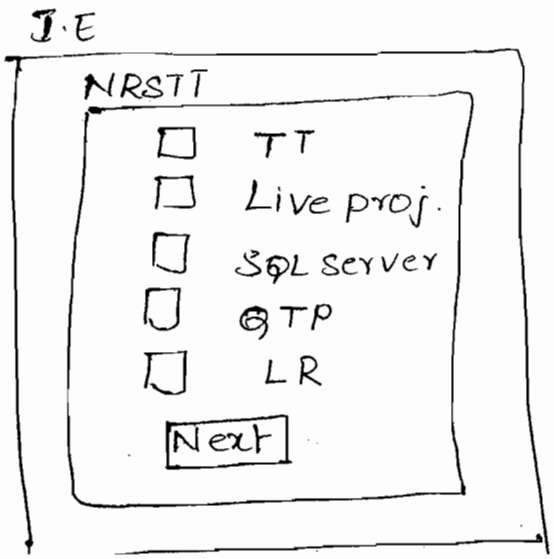
H.W

Case study - 1 (web)




```
Browser ("I.E"). page ("customer") . webEdit ("cID").
                                        set "007" ,
Browser("I.E"). page ("customer" / ~~c ph NO~~ ). webEdit("c ~~Add~~ phNO") .
                                        set "1111111111"
Browser("I.E"). page ("customer" / ~~cAdd~~ ). webEdit ("cAdd") .
                                        set "Hyd"
Browser("I.E"). page ("customer"). webEdit ("CcarNO").
                                        set "1234"
Browser("I.E") . page ("customer"). webButton ("submit").
                                        click.
```

Case study - 2 (web)



Develop the script for selecting TT check box, L proj., SQL server check box & clicking on Next button.

Browser("I.E").page("NRSTT").webcheckbox("TT").
set "ON"

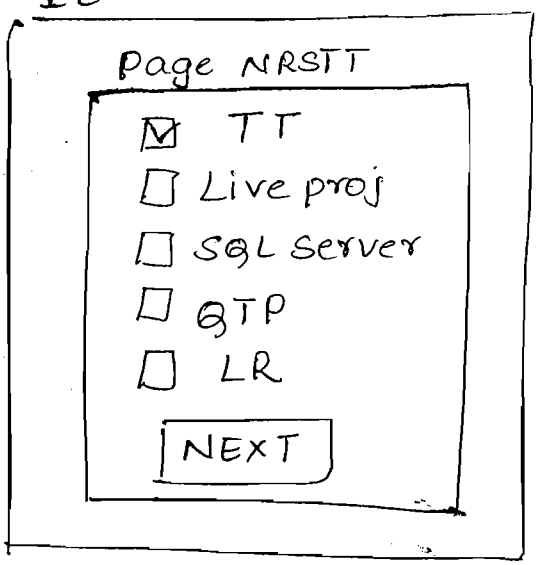
Browser("I.E").page("NRSTT").webcheckbox("Livepr").
set "ON"

Browser("I.E").page("NRSTT").webcheckbox("SQLserver").
set "ON"

~~Browser("I.E").page("NRSTT").webcheckbox(~~

Browser("I.E").page("NRSTT").webButton("Next").
click

Case study-3



Dev. The script deselecting the TT checkbox, selecting the live proj., QTP & LR checkboxes & clicking on Next button.

Browser("I.E").page("NRSTT").webcheckbox("TT").
set "OFF"

Browser("I.E").page("NRSTT").webcheckbox("Liveproj").
set "ON"

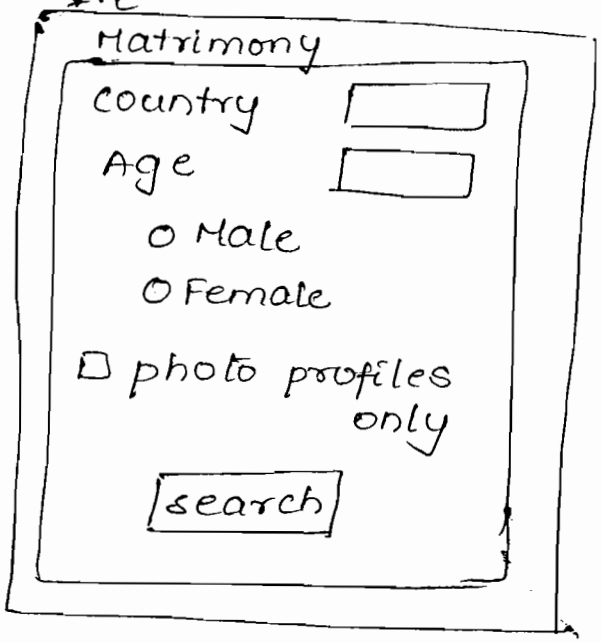
~~Browser("I.E").page("NRSTT").webcheckbox("SQLproj").~~
~~set "ON"~~

Browser("I.E").page("NRSTT").webcheckbox("QTP").
set "ON"

Browser("I.E").page("NRSTT").webcheckbox("LR").
set "ON"

Browser("I.E").page("NRSTT").webButton("NEXT").
click

Case study - 4

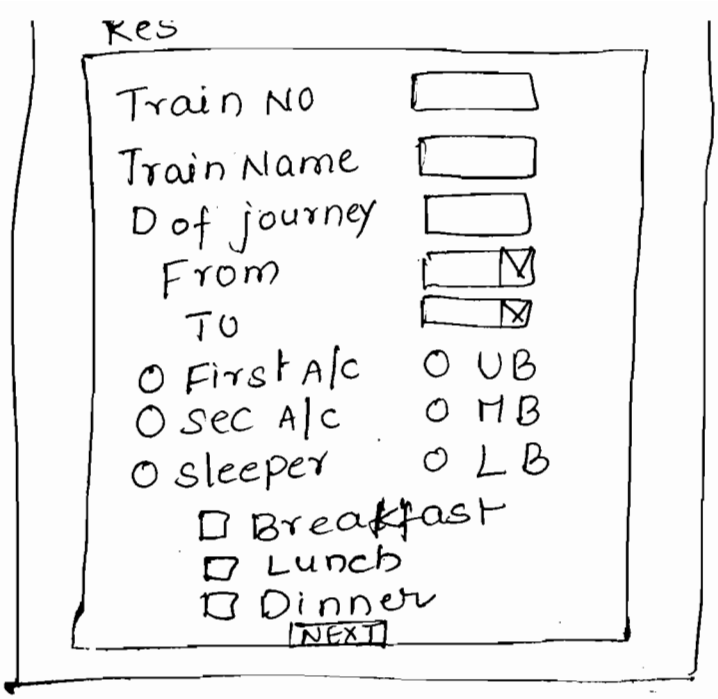


Dev. The script for entering, country, Age, selecting the gender, selecting the checkbox photo profiles only & clicking on search button.

```
Browser("I.E").page("matrimony").webEdit("country").
    ~~webEdit("Age")~~                    set "INDIA"
Browser("I.E").page("matrimony").webEdit("Age").
                                          set "28"
Browser("I.E").page("matrimony").webRadiogroup
                                ("Male").set
Browser("I.E").page("matrimony").webcheckbox
                                ("ph.profiles only").
                                          set "ON"
```

Case study - 5

Develop the script for entering Train NO., Train Name, Date of Journey, selecting the starting & Destination points, selected the reqd. class, selecting the req. berth, selecting the BF, Dinner checkboxes and clicking on NEXT.




```
Browser("I.E").page("Res").webEdit("Train NO").
                                    set "1234"
Browser("I.E").page("Res").webEdit("Train Name").
                                    set "Falaknuma"
Browser("I.E").page("Res").webEdit("D of journey")
                                    set "19/2/2009"
Browser("I.E").page("Res").weblist
```

Get RO property :-

Get RO property is used for getting the runtime object's property's value during the execution.

```
var = obj.heirarchy.getroproperty ("prop.name")
```

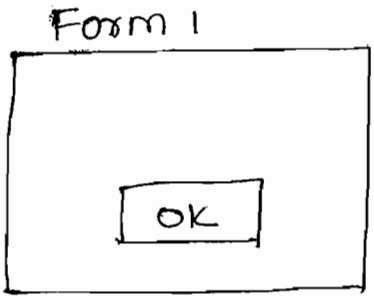
Get to property :-

Get to property is used for getting the test object's property's value during the execution.

```
var = obj.heirarchy.Gettoproperty ("prop.name")
```

Case study - 6

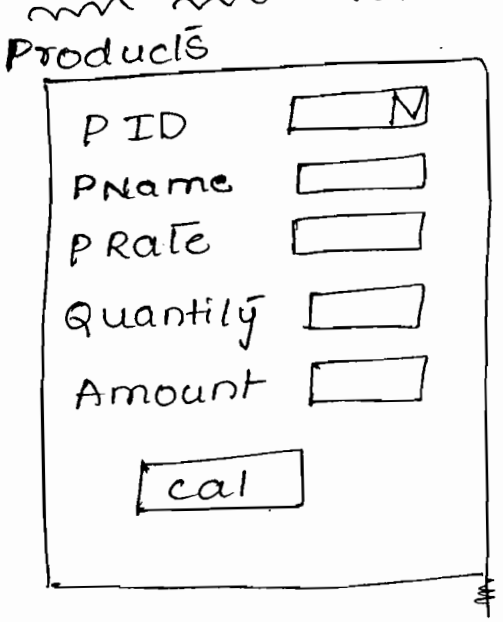
Develop the script for clicking on OK button if it is existing and enabled otherwise corresponding msg should be shown in the results.




```
var = vbwindow (" Form 1"). Vb Button("OK").exist
  If (var = True) then
    var 1 = vbwindow (" Form 1"). vbButton("OK").
                    getroproperty ("enabled")
      If (var1 = True) then
       vbwindow ("Form 1"). vbbutton ("OK").click
  else else
```

else
Reporter.Reportevent 3, "my rep", "ok button doesnot exist"
End if

Case study-7



Develop the script for selecting the PID, entering the Quantity, clicking on cal button & checking whether cal button is working properly or not.

→ PID automatically PName, PRate comes.

```
'Vbwindow ("products").Vbcombobox ("PID").
                                    Select "007"
q = 6
Vbwindow ("products").VbEdit ("Quantity").
                                    Set "q"
r = Vbwindow ("products").VbEdit ("pRate").
                        getropropelty ("text").

EV = r x q
Vbwindow ("products").VbButton ("cal").click
AV = Vbwindow ("products").VbEdit ("Amount").
                        getropropelty ("text")
If EV = AV Then
reporter.report event 0, "my rep" "cal button
                                is working propely"
else
reporter.reportevent 1, "my rep", "cal button is not
                                working prop"
```

H.W Case study 8 :-

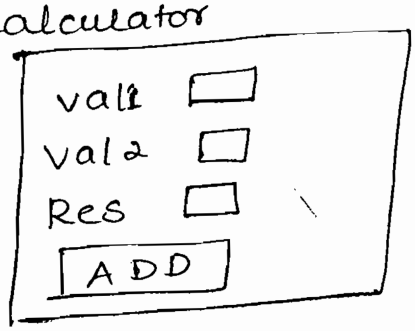
Develop the script for selecting the PId, entering the quantity, clicking on cal button if it is existing & enable and checking whether it is working fine or not.

```
Vbwindow("products"). vbcombobox ("PID").select "007"
q = 8
Vbwindow("products"). vbedit("Amount").set q
r = vbwindow ("products"). vbedit("prate").
                          getroproperty("Text")

Ev = q x r
var = vbwindow ("products"). vb Button ("cal").exist
If (var = True) Then
      var1 = vbwindow ("products"). vbButton("cal").
                          getroprop ("enabled")
   If (var1 = True) Then
     vbwindow ("products") . VbButton ("cal").
                                            click
     AV = vbwindow ("products") . vbedit("amt").
                          getroproperty ("Text")
       If EV = AV Then
         Reporter.Reportevent 2, "my rep",
                              "cal button is
       else                    working properly"
         Reporter.Reportevent 1, "my rep",
                  "cal button is not working
       End If                  properly"
   else
   Reporter.reportevent 3, "my rep", "cal
                button is not found"
```

```
Reporter. Reportevent 3, "my rep", "cal button
                                  doesnot exist"

End if
```

Case study - 9

calculator



Develop the script for entering 10 into val1 field, 20 into val2 field, clicking on Add button if it is existing & enabled & checking whether it is working properly or not.

```
systemutil.Run "c:\Documents and settings\shashi\
                                Desktop\calc.exe"

Vbwindow("Form1").VbEdit("val1").set "10"
Vbwindow("Form2").VbEdit("val2").set "20"
Var = vbwindow("Form1").VbButton("ADD").Exist
  If (Var=True) Then
 Var1=Vbwindow("Form1").VbButton("ADD").
                      Getroproperty("Enabled")
    If (Var1 = True) Then
      Vbwindow("Form1").VbButton("Add").click
       AV = Vbwindow("Form1").vbEdit("res").
                      Getroproperty("Text")
        If (AV=30) Then
          reporter.reportevent 0,"my rep",
                  "Add Button is working properly"
        else
         reporter.reportevent 1,"my rep","Add
                  Button is not working
                                     properly"
        End if
   else
```

```
        reporter.reportevent 2, "my rep", "Add Button
                                          is not enabled"
    End if
else
reporter.reportevent 3, "my rep", "Add Button
                                   doesnot exist"
End if
```

```
EV = 30
If (CINT(EV) = CINT(AV)) then
```

button Set to property objects thru scripting?
"

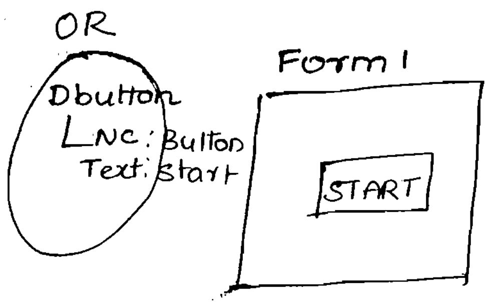
Set to property is used for setting the test object's property value temporarily during the execution.

ton
st"

Syntax

Obj. heirarchy . Settoproperty "prop.name", "prop.value"

Case study - 10



Develop the script for clicking on a button for 3 times whose text property value is been dynamically changing from start to stop and stop and start.

```
Vbwindow("Form1").VbButton("Dbutton").click
Vbwindow("Form1").VbButton("Dbutton").
                settoproperty "Text", "stop"
Vbwindow("Form1").vbButton("Dbutton").click
Vbwindow("Form1").VbButton("Dbutton").
                settopropelty "Text", "start"
Vbwindow("Form1").VbButton("Dbutton").click
```

~~setto~~

Simple Vbscript fns used during comparisions and calculations

1) CINT

It is used for converting any value into integer

2) CSTR

It is used for converting any value into string

3) cdbl

It is used for converting any value to decimal.

4) cbool

It is used for converting a value to boolean. (True or False)

5) LTrim : It is used for trimming left side spaces.

6) RTrim : It is used for trimming rt. side spaces

7) Trim : It is used for trimming the both side spaces.

8) Left : It is used for getting the left side substring.

Syntax

Var = Left(string, NO. of characters)

EX

Var = SURESH

Var1 = left(var, 4) ⟶ (sure)

9) Right

It is used for getting the right side substring.

EX

Var3 = right(var3, 3) ⟶ (esH)

10) Mid

It is used for getting the middle substring.

syntax

var = mid(string, starting character no., no. of characters)

EX

Var4 = mid(Var, 2, 4) ⟶ URES

11) Len

It is used for getting the length of a string.

12) Ucase

It is used for converting a string into uppercase

13) Lcase :

It is used for converting a string into Lowercase.

14) Round :

It is used for rounding a decimal value to the nearest integer.

Datatable Methods

while execute QTP uses run time datatable

Global sheet — Common info. for all actions.

Global — automatically iterates

local — no

Datatable methods are used for performing the operations on the runtime datatable

1. Addsheet : This method is used for adding an extrasheet to run time datatable

syntax datatable. Addsheet "sheetname"

2. Deletesheet :

This method is used for deleting a specified sheet from run time datatable.

syntax datatable. Deletesheet "sheetname"

3. Import

This method is used for importing the data present in xl file to the runtime datatable.

4. Import sheet

This method is used for importing a specified sheet of data from the xl file to a specified sheet in the run time datatable.

Syntax datatable . importsheet "path of the xl file", source sheet Id, destination sheet Id.

* xl file directly compatable with @TP

5. Export

This method is used for exporting the ~~spe~~ data present in the runtime datatable to xl file in a specified location.

Syntax Datatable. export "path of the location with a file name . xls extension"

6. Export sheet

This is used for exporting a specified sheet of data from the run time datatable to a specified location.

Syntax datatable . exportsheet "path of the xl file", sheet id to be exported

7. set current Row

This method is used for making the @TP focus on a specified row.

Syntax:

datatable. setcurrentrow ("row no.")

8. Set nextrow

This method is used for making the GTP focus on the next row of the currently focussed row.

Syntax

datatable.setnextrow

9. setprevrow

This method is used for making the GTP focus on the previous row of the currently focussed row.

syntax

datatable.setprevrow

10. Value method

This method is used for getting a value from a specified sheet, specified column and currently focussed row.

Note: This method is an optional method

syntax: datatable.value ("col.name", sheet Id)

** Imp

*11. Get sheet

This method is used for making the GTP focus on a specified sheet.

Syntax datatable.Getsheet ("Sheet id)

12. Get row count.

This method is used for getting the row count of a specified sheet.

Note By default it will get the rowcount

of specified sheet Then we need to first focus that sheet and Then get the row count.

rently

Syntax

1. Var = datatable . Getrowcount
2. var = datatable . Getsheet (sheet id). Getrowcount

Ex

```
datatable . Addsheet "suresh"
datatable . importsheet "d:\fl_auto\td\td1.xls", 1, 3
n = datatable . getsheet (3) . Getrowcount
For i = 1 to n
    datatable . setcurrent Row (i)
Vbwindow ("cal"). vbEdit ("val1") . setdatatable.value ("v1", 3)
    "       "       "    ("val2"). setdatatable("v2",3)
    "       "       vbButton ("Add") . click
expval = datatable("ev", 3)
actual = vbwindow ("cal") . VbEdit (" Result").
                GetRoproperty (" text")
if (cint (expval) = cint (actual) Then
    datatable ("res", 3) = "pass"
    else
    datatable ("res", 3) = "Fail"
End if
Next
datatable . Exportsheet "d:\fl_auto\log\file2.xls", 3
datatable . Deletesheet "suresh"
```

Input output parameters

Checkpt. statement Internal file — exeeu time
o/pvalue " incr

else getro

Input output parameters is a concept introduced by QTP which is used for passing some values to the action while calling it and returning some values from the action.

Navigation for declaring the input & output parameters

Active the menu item 'Edit'

↓

Goto action

↓

Select the option 'Action prop.'

↓

Select the parameters tab

↓

Add the reqd. no. of input & o/p paramt.

↓

click on ok

After declaring the i/p & o/p paramt we need to use in test with foll. syntax.

syntax parameter ("name")

make action reusuable

```
Vbwindow ("cal").vbEdit ("val1") . set parameter ("a")
   "        "        " ("val2") .        "       ("b")
   "        "        " ("Add") . click
var1 = vbwindow("cal") . VbEdit ("Res") . Getroproperty ("Txt")
    parameter ("c") = var1
vbwindow ("cal") . vbEdit ("sub") . click
var2 = vbwindow ("cal") . vbedit ("Res") . Getroproperty ("Text")
    parameter ("d") = var2
Vbwindow ("cal") . vbEdit ("Mul") . click
Var3 = Vbwindow ("cal") . vbEdit ("Res") . Getroproperty ("Text")
    parameter ("e") = var3
```

While calling the action, from any other test, we need to first specify all the i/p values and followed by the variables to catch the return values seperated by commas.

Ex:

```
Run Action "Action 1 [inout 133]", oneIteration, 10, 20, r1, r2, r3
```

(Run Action "Action 1 [inout 133]", oneIteration — call existing action)

```
msgbox r1
msgbox r2
msgbox r3
```

Note: If at all the input values are not specified then the default values will be used.

Automation Object Model (AOM)

AOM is a concept introduced inorder to perform the operations automatically on QTP through scripting.

Ex:

```
Dim qtapp
Dim qttest
set qtapp = createobject ("QuickTest.Application")
qtapp.launch
qtapp.visible = true

qtapp.open "d:\aom1", true
set qttest = qtapp.test
qttest.run
qttest.close

qtapp.open "d:\aom2", true
set qttest = qtapp.Test
qttest.run
qttest.close

qtapp.quit
```

class name

Diff. Action & fn

Database connections

databases
flat files → Testdata — xl files

improve performance
user friendly & compatable

Db
↓
Tables
↓
record.

activex obj.

3 party softare — drivers

SQL, or — store in seperate server not into files

Database connection for microsoft access:

```
Dim con, rs
set con = createobject("adodb.connection")
set rs = createobject("adodb.recordset")

con.provider = "microsoft.jet.oledb.4.0"
con.open "d:\testdata.mdb"

rs.open "select * from info", con

Do while not rs.eof
Vbwindow("cal").vbEdit("value1").set rs.fields("v1")

Vbwindow("cal").vbEdit("value2").set rs.fields("v2")

Vbwindow("cal").vbButton("Add").click
   rs.movenext
Loop
```

Db connection for oracle

obj linking embedding

```
con.open "provider = oraoledb.1; server = localhost;
          uid = scott; pwd = tiger;
          database = testdata"
```

Db connection for SQL server

```
con.open = "provider = sqloledb.1; server = localhost;
            uid = scott; pwd = tiger;
            database = testdata"
```

Dynamic handling of Object Repositories

During execution,
Shared rep.

One can handle the object repositories dynamically like adding them to the actions, removing them from the actions with the help of a utility object by name repositories collection.

Ex:

```
reppath = "d:\sr123.tsr"
repositoriescollection.Removeall
repositoriescollection.Add(reppath)
vbwindow("cal").VbEdit("val1").set"10"
vbwindow("cal").VbEdit("val2").set"20"
vbwindow("cal").vbButton("Add").click
Var = repositoriescollection.Find(reppath)
repositoriescollection.Remove(var)
```

Descriptive programming

If at all the object's description is present in the program itself then that type of programming is known as descriptive programming

(By obj spy)

1st style of DP :

```
Dialog ("text:=Login", "width:=320").Activate
Dialog ("text:=Login").winEdit ("attachedtext:=
        Agent Name:").set "suresh"
Dialog ("text:=Login").winEdit ("attachedtext:=
        password:").setsecure - - -
Dialog ("text:=Login").winButton ("text=OK").click
```

2nd style of DP

```
set lin = description.create
lin("text").value = "Login"
lin("width").value = "320"

set an = description.create
an("attachedtext").value = "Agent Name:"

set pw = description.create
pw("attachedtext").value = "password:"

set ok = description.create
ok("text").value = "OK"

Dialog(lin).Activate
Dialog(lin).winEdit(an).set "suresh"
Dialog(lin).winEdit(pw).setsecure - - -
Dialog(lin).winbutton(ok).click
```

3rd style of DP

1. Define all the reqd. objects' description in the library file.
2. Assign that library file to the reqd. test.
3. Use those objects in the script as follows:

```
Dialog (lin) . activate
Dialog (lin) . winEdit (an) . set "suresh"
Dialog (lin) . winEdit (pw) . setsucure . . .
Dialog (lin) . winButton (OK) . click .
```

OR

1) Time consuming

Easy to develop & maintain
limited time
incase of project

DP

more time.
In future
Lifelong testing execution is fast.

QTP defect

sometimes cannot identify obj. through propu info available in OR.

In this case DP is useful.

Dynamic handling of file system

log file means high level results file.

1 - read mode
2 - write mode
8 - append mode

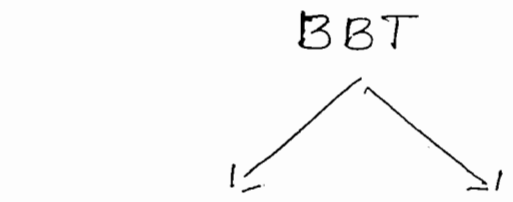
True - visible

One can perform the operations on the file system by creating an object for "scripting.filesystemobject" class as follows :-

Ex

```
Set fso = createobject ("scripting.filesystemobject")
Set myfile = fso.opentextfile ("d:\logfile133.txt",
                                8, true)
Vbwindow ("cal").VbEdit ("value1").set 10
Vbwindow ("cal").VbEdit ("value2").set 20
Vbwindow ("cal").VbButton ("Add").click
Var = Vbwindow ("cal").VbEdit ("Result").
                        getroproperty ("text")
myfile.writeline ("The result is" & var)
myfile.close
```

Load Runner



Fund — perfor

SLAS
service level Agreement - accepted reqts.

PERFORMANCE TESTING USING LOAD RUNNER

There are 2 catagories in black box Testing.

1. Functional Testing
2. Non Functional Testing.

Functional Testing

In This catagory, one will concentrate on the functionality of all the features of the application.

Non-Functional Testing

In this catagory, one will concentrate on the following areas:

2. Installation "
3. volume "
4. usability "
5. performance " ← E

* load = No. of users

Load Testing:

It is a type of testing in which one will apply initial load on the system and sequencially increases the load inorder to find the critical load. once the critical load is found, it will be compared with the target load.

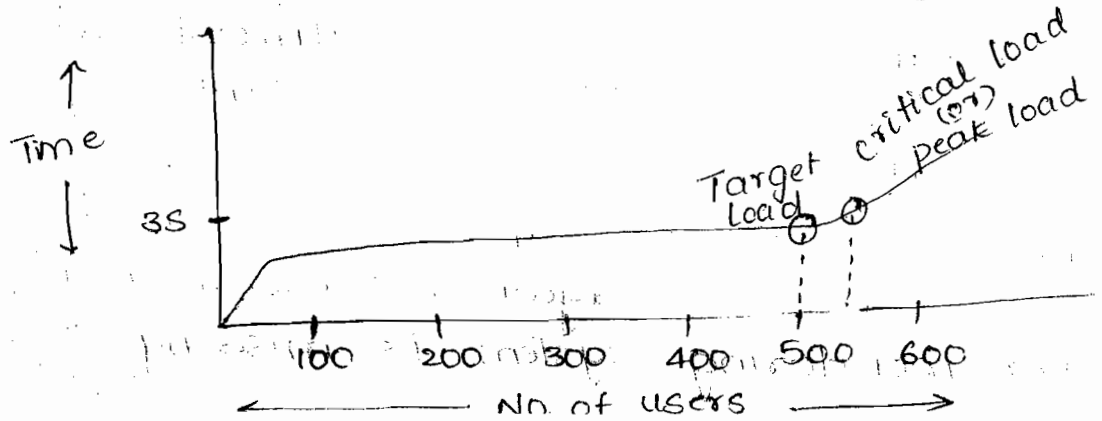
If at all the critical load is ≥ Target load, then one will come to a conclusion that the load testing is passed otherwise failed.

Target load:

The customer expected load is known as target load.

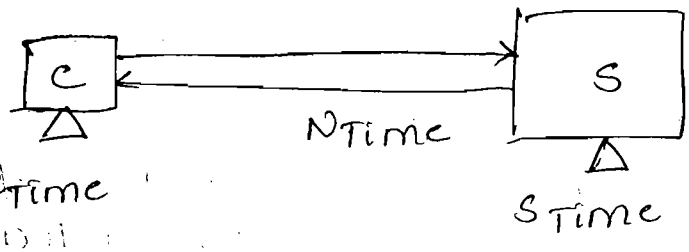
critical load (or) peak load:

The load beyond which application starts degrading its performance is known as critical load.



Features performance Testing :-

It is a type of testing in which one will apply predefined quantified request & measured calculate the response time. If at all the response time is ≤ expected time Then one will come to a conclusion that the features performance testing is passed. Otherwise failed.



$R_{time} = C_{Time} + Network_{Time} + Server_{Time}$

Stress testing :-

It is a type of testing in which one will perform some abnormal actions intentionally on the application or use the application continuously for long period of time inorder to check the stability of the application.

Drawbacks of Manual Load Testing

1. More no. of human resources are reqd.
2. simultaneous actions are almost impossible.
3. coordination & cooperation is missing.
4. cannot repeat the same task again & again in same fashion.

LoadRunner

Introduction

Type of the tool : performance tool

company : Introduced by Mercury Interactive Incorporation & ~~recently~~ right now taken over by HP.

Version : 8.0 (matured version)

Latest 9.0

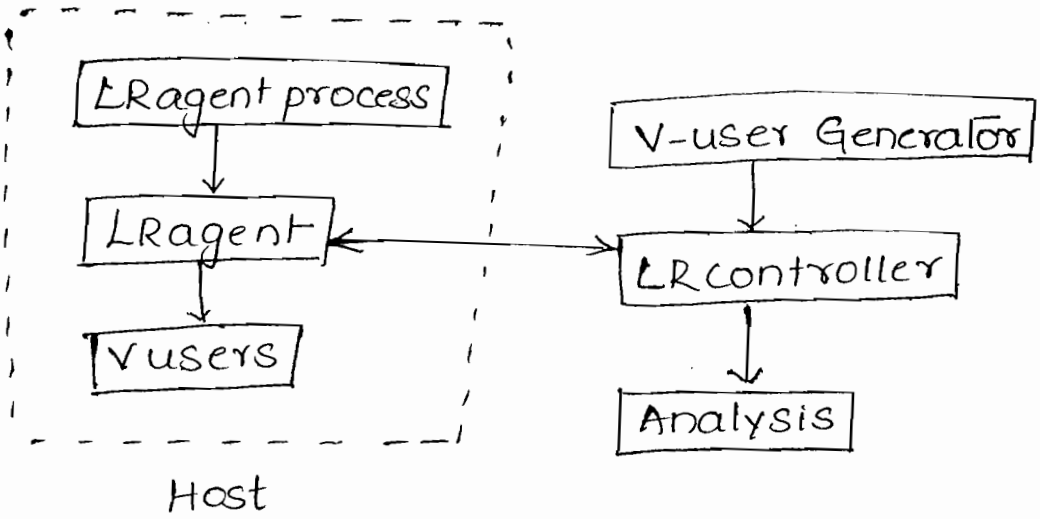
scripting language : vuser script

Advantages of LoadRunner :

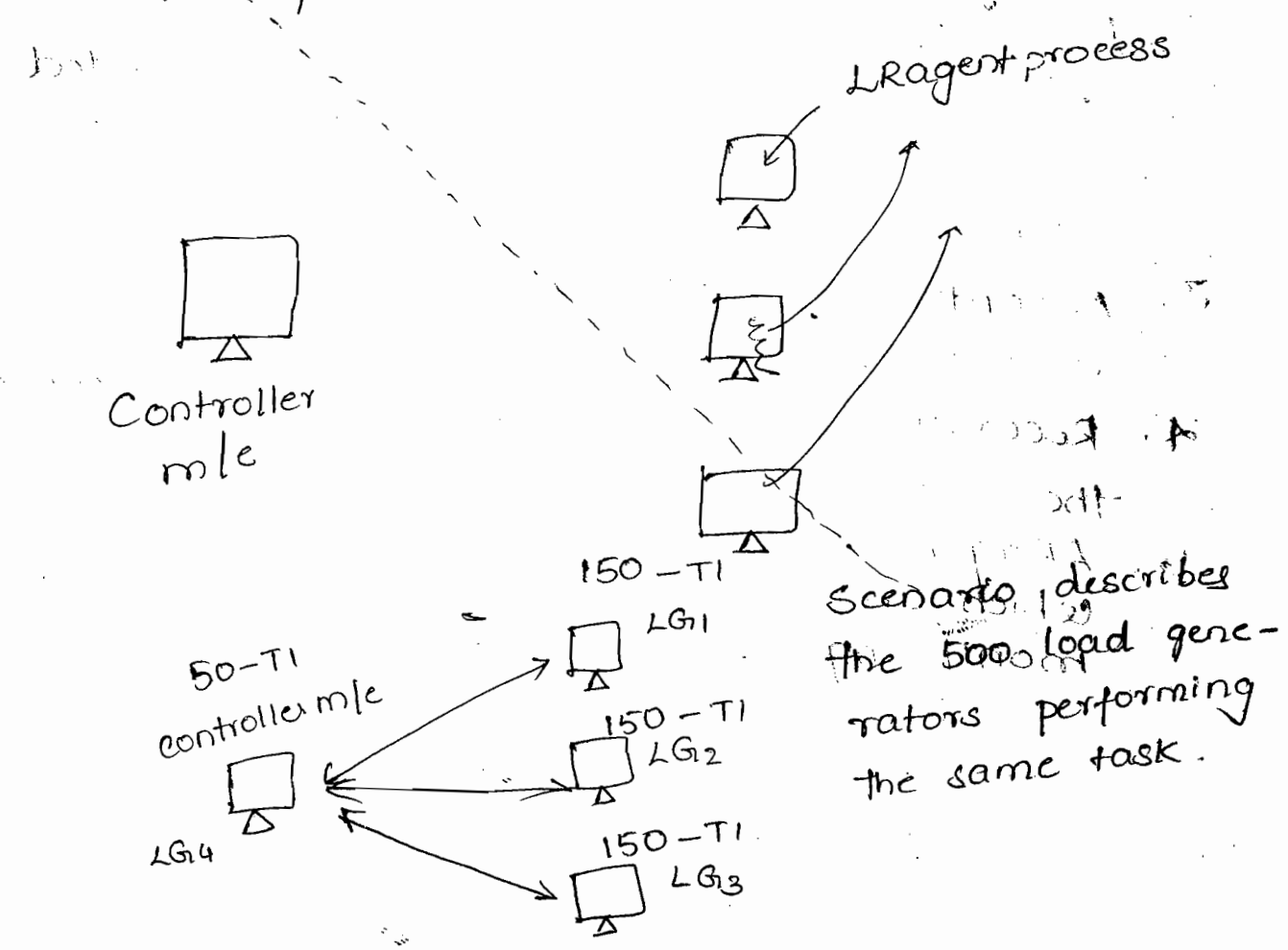
1. Vusers concept overcomes the more no. the human resources problem.
2. Rendezvous point is a feature provided (Randevo) by LR which overcomes the simultaneous actions problem.
3. LRcontroller overcome the coordination and cooperation problem.
4. Recording and playback facility overcome the repitition problem.
5. LRagent is a feature provided by LR which overcomes the problem of proper monitoring.

**V. Imp

Loadrunner Architecture



4 Components



V-user generator

Virtual user generator is a 1st and foremost component of Load runner which is used for the following:

1. Generating the basic script
2. Enhancing the script
3. checking whether the script is running properly or not by analysing the results.
4. saving the scripts in terms of a task.

LR controller

LR controller is a vital component of Loadrunner which is used for the following:

1. It is used for controlling the whole load testing process.
2. It is used for designing the scenarios.
3. " " executing " " .

Scenario

Scenario is defined as a situational package which describes how many number of vusers need to be generated for which machines and perform what tasks for how much duration.

(or)

Scenario is defined as a simulated real time situation.

LR agent process

LR agent process recieves a remote command from the LR controller and launches the LR agent whenever reqd. in the load generator.

LR agent

LR agent is a vital component which is used for the foll:

1. It is used for generating the reqd. no. of vusers.
2. It is used for starting the vusers.
3. while running it will monitor the v-users.
4. while monitoring, it will record the individual information of each and every vuser.
5. Once the work is done, it will kill the v-users.
6. Finally it will consolidate the individual info. and send it to the LR controller.

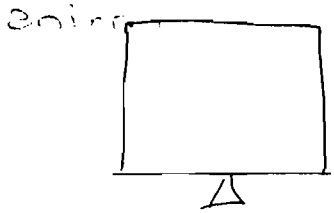
Note:

The LR controller will recieve the raw results from all the LR agents in the load generators and will send it to the analysis.

Analysis:

The analysis component will convert the raw results into understandable format so that

Load generator m/e should have LR agent process.



Controller m/e should have
V-user generator
LR controller
LR agent process

Load gen.
LR agent process

analysis

Loadrunner Life cycle

It contains 5 phases.

1) Load Test planning
2) creating the tasks
3) Designing the scenarios
4) Executing the scenarios
5) Analysing the results

Load test planning

In this phase, the load test lead will do the foll:-

1. He will identify the components of the system (load testing lab).
2. He will analyse the configuration of the components.
3. He will calculate the response time.
4. He will analyse how much load can be generated from each m/e.

5. He will analyse peak timings & off peak timings of the appl.

6. He will analyse the climatic conditions if reqd.

7. He will analyse how many scenarios need to be designed.

8. He will do the resource planning.

9. He will do the scheduleing.

10. He will prepare load test plan doc based on the above analysis.

Creating the tasks

In this phase, one will do the foll. with the help of virtual user generator.

1. generating the basic script.
2. Enhancing the script
3. checking whether the script is working properly or not.
4. saving the scripts interms of tasks.

Virtual user Generator

~~Virtual user generator is~~

while opening a new test in v-user generator, we need to select the desired protocols inorder to make the loadrunner understand about that environment.

Once the new test is opened, there will be 3 blocks in which we can generate

1. V-user init block
2. Action block
3. v-user end block

V-user init

Whenever some task need to be performed only once in the starting then that task related script need to be kept in the v-user init block.

Action

Whenever some task need to be performed repeatedly continuously for some duration then that task related script need to be kept in action.

V-user end

Whenever some task need to be performed only once at the end of the task then that related script need to be kept in the V-user end.

Note:

While running the loadrunner will perform the actions directly on the business logic in the appl. server but not on the front end appl. present in the browser.

Adv → Identification time is avoided.

Enhancements in Vuser scripts

To identify bottle necks in a software processes, ~~execution speed~~ the corresponding testers are enhancing Vuser scripts using transactions, Rendezvous points, Vuser script runtime settings etc.

Transaction point

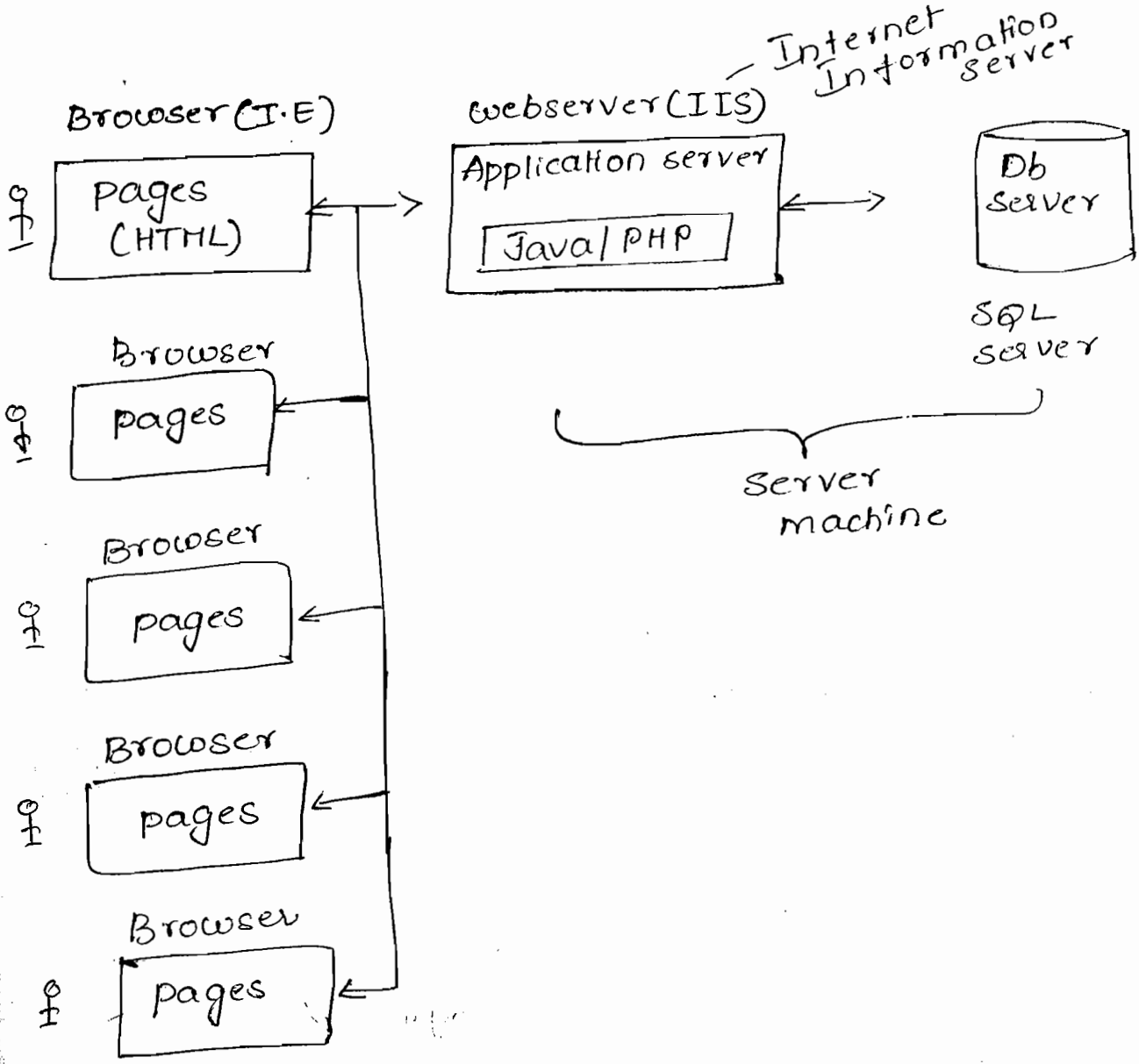
We can use this concept to mark reqd. feature related code in Vuser script. If u mark any feature code as transaction, then the loadrunner is returning performance results for that transaction individually.

```
Vuser_init()
{
  ===
}
Vusers
Action()
{
  ===
  lr_start_transaction("name");
  ===
  lr_end_transaction("name", flag);
  ===
}
Vuser_end()
{
  ===
}
```

HTTP
Hypertext transfer protocol

Html
Hyper Text markup language

Ex: Mercury Tours website



Navigation to insert transaction

select position in Vuser script action

↓

Insert menu

↓

start transaction

↓

enter transaction name as 'Login'

↓

Click OK

↓

Click OK

Select position at the end of the reqd. code

↓

Insert

↓

end transaction

↓

click OK

Rendezvous point

To maintain load balancing during script execution under load, the testers are using this point in vuser script.

```
vuser_init()
{
=======
}
Action()
{
=======
=======
=======
lr_Rendezvous("name");
lr_start_transaction("login");
=======
=======
lr_end_transaction("login", LR_Auto);
=======
}
vuser_end()
{
=======
=======
}
```

Navigation

Select position on The top of transaction
↓
Insert
↓
Rendezvous
↓
Enter Name
↓
Click OK

{ arguments

LAST); ← means arg. are finished.

Checkpoints :-

There are 2 checkpoints in Loadrunner.

1. Text checkpoint
2. Image checkpoint

Text checkpoint :

Text checkpoint is used for checking whether The proper text is been responded by The server under The load or not.

We can insert the text checkpoint during recording also.

Syntax

web_reg_find ("Text = Expected Text", LAST);

~~We c~~

We can insert the text checkpoint manually after rec. also

Navigation:

Keep the cursor in the desired locate

↓

Activate menu item 'new step'

↓

Expand the web checks

↓

Select the option 'Text check'

↓

click on OK

↓

Specify the expected text

↓

If reqd.; specify the text present in the right of it & left of it

↓

click on OK

To avoid the above navigation, one can directly insert the foll. statement wherever reqd.

Syntax

```
web_find ("web_find", "right of = Text",
          "left of = Text", "what = expected
                              text", LAST);
```

```
web_find ("web_find", "Right of = how r u",
          "Left of = hai", "what = hello", LAST);
```

Image checkpoint :

Image checkpoint is used for checking whether the proper image is been responded by the server under the load or not.

Navigation

Keep the cursor under desired location
↓
Activate, menu item 'Insert'
↓
Select the option 'New step'
↓
expand web checks
↓
Select the option 'Image check'
↓
click on OK
↓
Specify the alternative image name
(or)
Image server file name
(or)
Both of them
↓
click on OK.

To avoid the above navigation, one can directly insert the foll. statement:

Syntax

```
web_image_check("web_image_check",
    "src = server file name",
    "Alt = Alternative name", LAST);
```

Ex :

```
web_image_check("web_image_check",
    "src = serverimage123", "Alt = suresh", LAST);
```

Transaction points are used for calculating the time taken by the server to respond for any user request. (It will calculate the complete response time for every user request)

Syntax & Navigation } as discussed earlier.

* Rendezvous point = Meeting point

is used for making all the users meet at one point & releasing it at a time.

Comments :

comments are used for making the script more understandable.

Navigation

Keep the cursor in the desired location

Activate the menu item 'Insert'

↓

Select the option comment

↓

Specify the desired comment

↓

Click on OK.

Ex

```
/*
 * Image check will be done here
 */
```

```
/*
 * Hai
 * Hello
 * How r u
 * Bye
 */
```

Breakpoint

Togeling – ON/OF switch

~~Togeling~~

Breakpoint is used for breaking the execution temporarily.

Navigation

Keep the cursor in the desired location

↓

Activate the menu item 'Insert'

↓

Select option 'Toggle breakpoint'

Think time

Inorder to simulate the realtime scenario, just like real users think between the requests, if at all our v-users need to think then we need to insert think time statement wherever reqd.

Syntax :

```
Lr_think_time(time in sec);
```

Ex

```
Lr_think_time(10);
```

To replay the think time during running, we need to do the foll. settings :

Navigation

Activate the menu item v-user
↓
Select the option 'run time settings'
↓
Select the option 'think time'
↓
Select the option O replay think time.
↓
Select the reqd. option under it
↓
click on OK

P — To simulate with realtime scenarios.

Parameterization

It is a process of replacing the constant values with paramt. or variables inorder to increase the scope of the test.

Navigation :

Select the const. value in the script
↓
rt. click on it
↓
Select 'replace with parameter'
↓
Specify the parameter name
↓
click on OK
↓
Select the parameter, rt. click on it
↓
select the option 'parameter prop.'
↓
Click on edit with Notepad
↓
Specify all the values
↓
Save the file & close
↓
Specify the reqd. col. settings
↓
Specify how the next row need to be selected
↓
Specify when the value need to be updated
↓
click on close

Navigation 2

Activate the menu item Vuser

↓

Select the option 'paramcter list'

↓

Click on New

↓

Specify the paramt. name

↓

click on 'create table'

↓

click on edit with notepad

↓

Specify all the values

↓

Save & close the file

↓

Specify the desired col. settings

↓

Specify how to select next row

↓

Specify 'update value on'

↓

close the paramcter list

↓

Select the desired const. value in the script
rt. click on it

↓

goto use existing parameters

↓

Select the desired parameter

Navigation for changing the run logic

Activate the menu item 'vuser'

↓

Select 'runtime settings'

↓

Select run logic

↓

Specify no. of iterations

↓

click ok

Navigation for changing the log settings

Activate the menu item 'vuser'

↓

Select the option 'runtime settings'

↓

Select 'log'

↓

select 'extended log' & reqd. settings

↓

click on ok

Correlation

Correlation is a process of capturing the dynamic data generated by the server and using it wherever reqd. in the script inorder to make the replay of the script successful.

Correlation can be done in 3 stages.

1. During recording
2. After executing the script once
3. Whenever automatic correlation fails then finally manually we do the correlation.

1. **Navigation for making the loadrunner to correlate automatically during recording:**

Activate the menu item 'Tools',
↓
Select the option 'rec options'
↓
Select the checkbox ☐ enable correlation during rec.
↓
click on OK

2. once the script is executed, then we can do the correlation in foll. way.

Activate the menu item 'Vuser'
↓
select the option 'scan action for correlations'
↓
Goto correlation results window

Then we need to go for manual correlation in the foll. way.

First we need to rec. 2 identical scripts

↓

we need to compare those scripts with

wdiff tool.

↓

windows

↓

Find the areas to be correlated

↓

Insert web_reg_save_param statements wherever required.

↓

Replace the dynamic data with the corresponding variable names.

Navigation for comparing the scripts

Activate the menu item tools

↓

compare with vuser

↓

Browse the desired script file

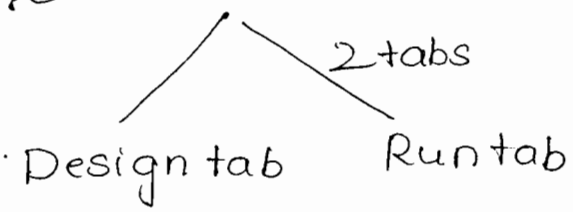
↓

click on open.

Syntax for web_reg_save_param :-

```
web_reg_save_param("variable Name",
    "lb = leftbound value", "rb = rightbound value",
    LAST);
```

lr Controller



lr. controller

lr controller is used for the foll:

1. Designing the scenarios
2. Executing the scenarios

Designing The scenarios

while designing a scenario, First we need to plan the no. of groups of v users, we need to assign the machines for those groups, we need to assign tasks for those groups and then edit the schedule.

Navigation

click of add Group
↓
Specify the Group name
↓
Specify the quantity of vusers
↓
specify the load generator name
↓
specify the task
↓
click on OK
↓
In the same manner, plan all the reqd.

Select one of the foll. options

1. Schedule by scenario
2. Schedule by group.

If at schedule by scenario is selected, do the foll. settings.

↓

Select the ramp up tab

↓

Select thone of the foll. options:

- O Load all vusers simultaneously
- O start [ ] Vusers every [ ] Time

↓

Select the duration tab

↓

Select one of the foll. options under it:

- O Run until completion
- O Run for [ ] Time
- O Run indefinitely

↓

Select the rampdown tab

↓

Select one of the foll options under it:

- O stop all vusers simultaneously
- O stop [ ] vusers every [ ] Time

If at all schedule by group option is selected. Then do the foll. settings for each group seperately

↓

Select start time tab and select one of the foll. options under it.

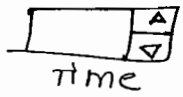
- O start at the beginning of the scenario
- O start [ ] after the scenario begins
- O start when group [ ] finishes.

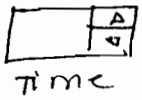
↓

Do the rampup, duration and rampdown setting as above for each group seperately

↓

click on scenario start time & select one of the foll. options:

- without delay
- with a delay of 

- At 

on 

↓

click on OK

↓

click on OK

↓

start scenario

---

no. of
hits/s requests given by vusers

throughput ~~serves~~ → ~~info~~ data sent by server/sec

is measured ~~by~~ in bytes & represents the amt. of data that vusers received from the server

Analysis

The analysis tool is used for analyzing the results with the help of reports and graphs.

Navigation for analyzing the results from Controller.

Activate the menu item 'Results'

↓

Select option 'Analyze results'

↓

once the report & graphs are analyzed, we need to prepare the load runner report & fill the conclusions in it and finally will send it to concerned dept.

Navigation for preparing reports

Activate menu item 'reports'

↓

select the option 'microsoft word report'

↓

Specify all the reqd. info

↓

click on ok