SOFTWARE TESTING :-

* Finding Quality of Software.
* Nature of testing is identifying hidden bugs in software.

Quality :-

* Monitoring
  Are we developing right software
* Measuring
  Developed s/w meeting with client requirement

WHAT TEST ENGINEER TEST ?

* Tester play every end user role to test software.
* Here tester verify developed s/w meeting with client requirement before shipment to client.

OBJECTIVE OF TESTING :-

* To ensure software is bug / defect free.
* To ensure correctness of software.
* To ensure completeness of software.
* To ensure robustness of software.
* To ensure reliability of software.

Scanned with CamScanner

* Early defects reduce cost of Defect fixing

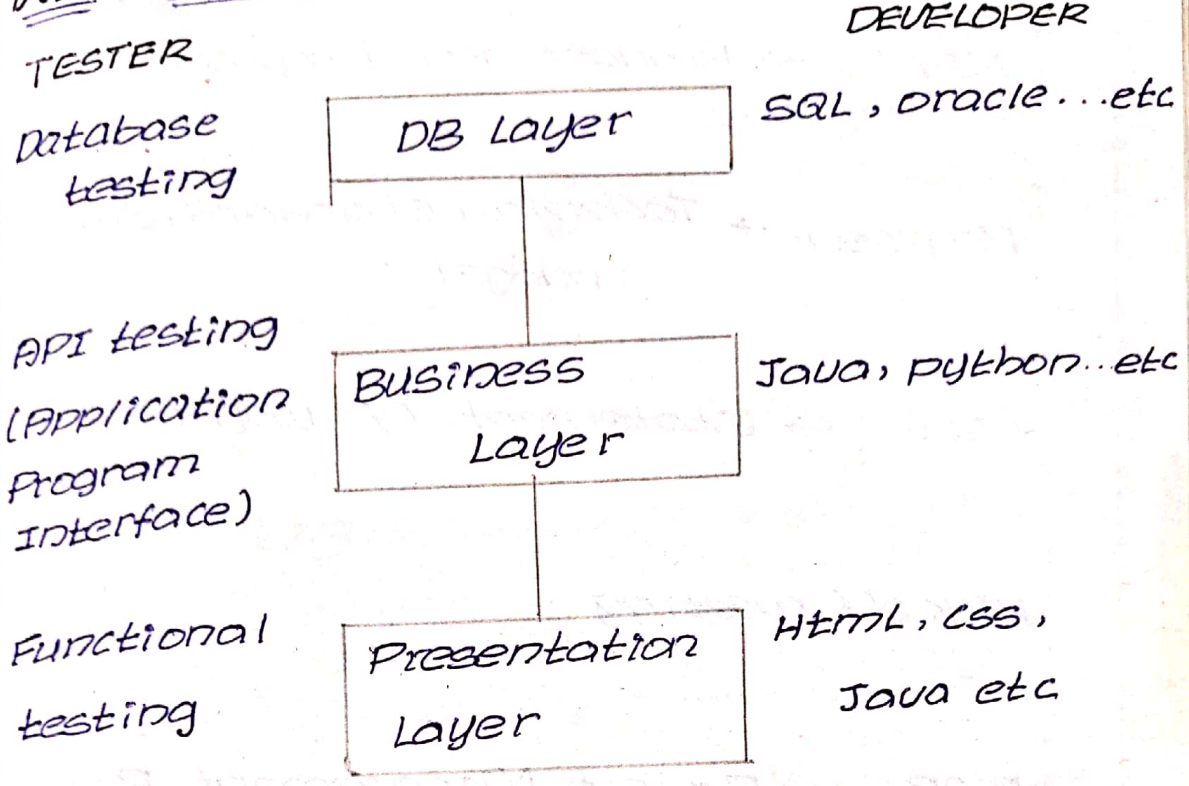
Requirement collection	0 %.
Requirement analysis	10 %.
Design	20 %.
coding	30 %.
	50 %.
Release	100 %.

THEORY :-

* Every program still there is a Error.
* When you are introducing something that means you are inventing new bug.

Scanned with CamScanner

WEB APPLICATION ARCHITECHTURE :



UI (User Interface) - AUTOMATION TOOLS :-

* Selenium
* QTP/UFT
* Rational
* Code UI
* RPA
* SOAP UI
* Tosea
* Katulon studio

Scanned with CamScanner

SOFTWARE BIDDING WORK :

RFP → Request For Proposal

Proposal → Technical standard, duration, budget

SOW → Statement Of Workorder

Kick off meeting

PMP → Project Management Plan

* **RFP :-**

client request to organization to send their proposals.

* **Proposal :-**

Every organization send proposal.

Proposal contains

* Achievements
* Technical standards
* Time
* Budget

Scanned with CamScanner

* SOW :-

Client select one of Proposal and release Statement of Workorder to particular organization.

* Kick off meeting :-

Organization organize lack of meeting with all employee to celebrate new project.

During this lack of meeting organization introduce project manager to company.

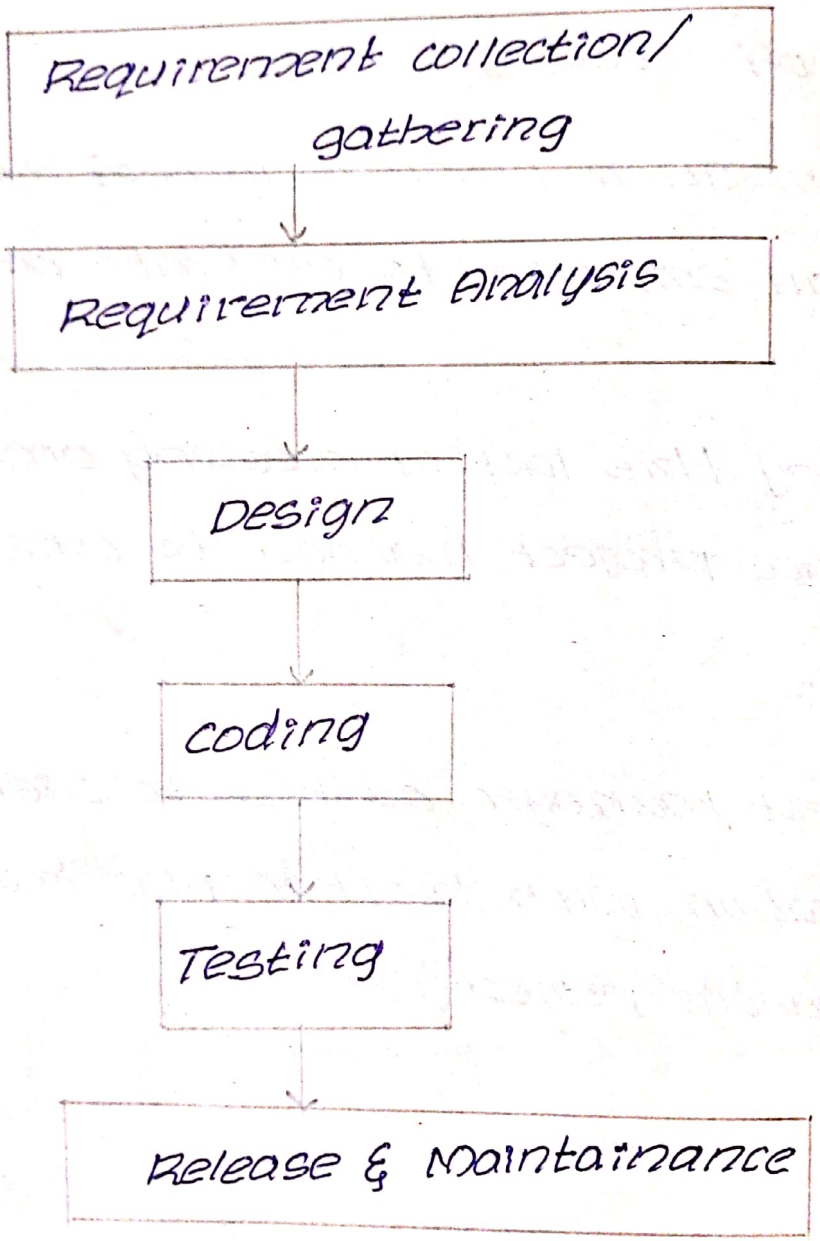
* PMP :-

Project manager address to client and organization with his/her plan how he/she can manage project.

Scanned with CamScanner

** SDLC (Software Development Life cycle) :-

A systematic approach to develop software from start to end.



Scanned with CamScanner

REQUIREMENT COLLECTION / GATHERING :-

* After completed bidding work organizat-ion sent Business Analyst team to meet stake holder (client).
* Within the meeting Business Analyst and client exchange their requirements.
* After completed requirement collection Business Analyst prepare few questions on requirements

1. Are they achievable requirements ?
2. Are they complete requirements ?
3. Are they correct requirements ?
4. Are they maintainable requirements ?
5. Are they testable requirements ?

* After completed reviews on client requirements business analyst finalize requirements and design BRS document.

BRS → Business Requirement Specification

CRS → Client Requirement Specification

URS → User Requirement Specification.

Scanned with CamScanner

WHAT IS BRS/CRS/URS DOCUMENT ?

* It contain real requirements of client.

* Before shipment slw to client, organ.
-ization and client team use same BRS document inorder to verify Quality of software.

* After completed Requirements gathering In agile testing product owner (BA) prepare user stories for Requirements.

WHAT IS USER STORY ?

A small narration what exactly client wants.

Inorder to write user story, Product owner follow unique design.

As a __ I want __ so that __

EXAMPLE :-

Flipkart

* As a shopowner I want to ADD my product to website

* As a customer I want purchase product from online.

Scanned with CamScanner

REQUIREMENT ANALYSIS :-

After completed review on BRS document system analyst conduct review on BRS to check feasibility of s/w requirements.

Feasibility :-

* Cost
* Budget
* Accepting terms and condition

* Once feasibility completed system analyst transfer all business requirement into s/w requirements.

* After completing system analyst list out all the s/w requirements in SRS document.

SRS → Software Requirement Specification

WHAT IS SRS DOCUMENT :-

* Technical information :
  - → Java
  - → Oracle
  - → J2EE
  - → Selenium

Scanned with CamScanner

* Functional information [FRS] :

→ Login

→ Registration

→ Add to cart

* Non Functional information [NFRS] :

→ 24/7 Action

→ security

→ performance capacity

→ storage

→ Access points

DESIGN :-

* HLD (High Level Design Document)

* LLD (Low Level Design Document)

After completed SRS document and their reviews. Designer categeory people (one of development team) start design of HLD and LLD diagrams.

HLD (High Level Design) :-

It explains overall architecture of software

1. System Architecture

2. Usecase diagram

Scanned with CamScanner

3. Database Architecture

4. Database design

LLD (Low Level Design):-

It explains internal structure of each HLD.

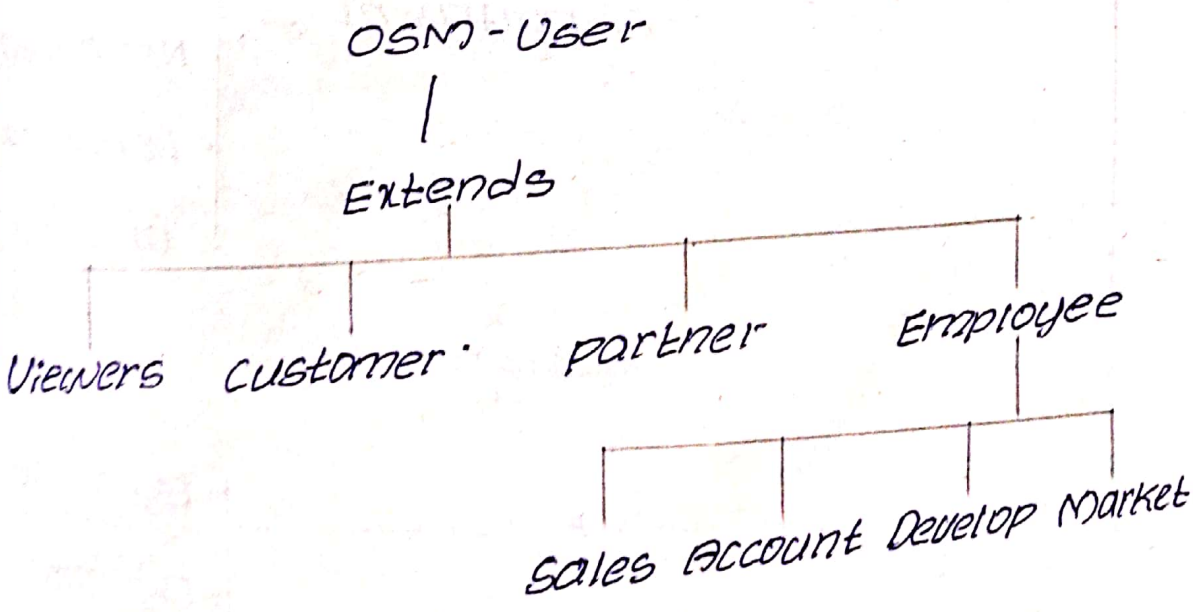
1. User model diagrams

2. Flow diagrams

3. Action diagrams

1. SYSTEM ARCHITECTURE:-

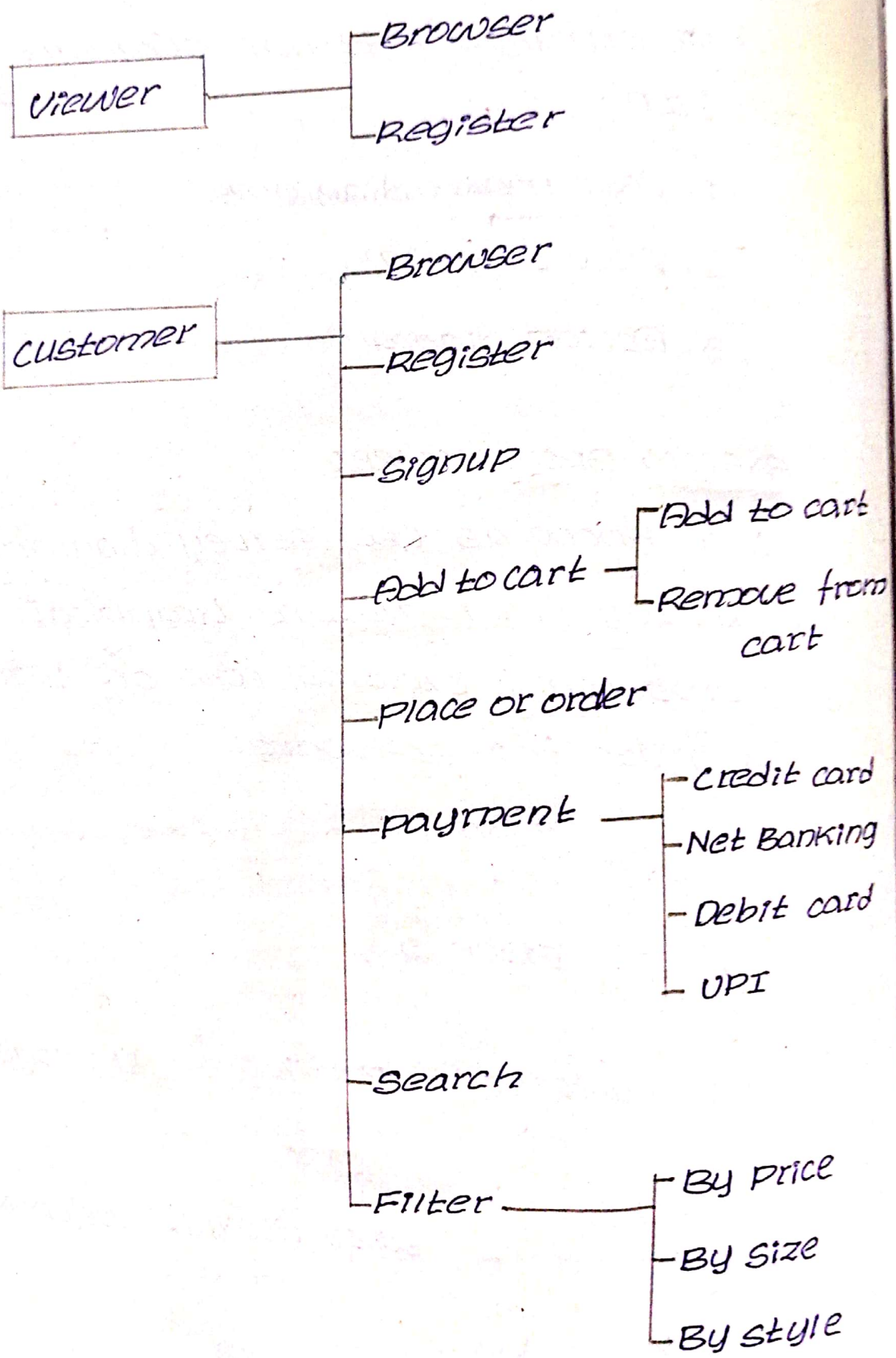
Also know as User servey diagram. User servey diagrams are Graphical diagrams which explains who are working users under the software.



Scanned with CamScanner

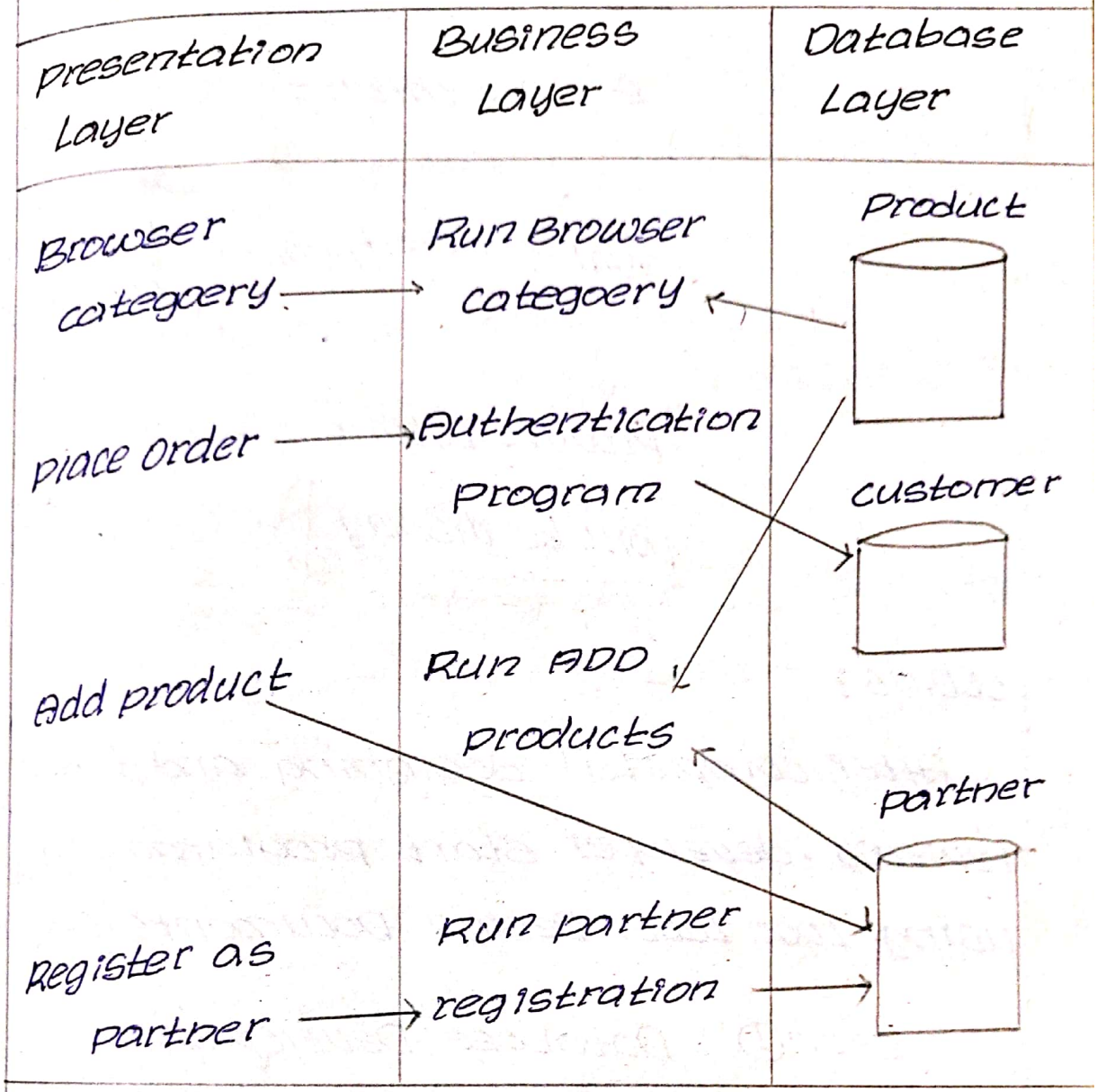
2. USER DIAGRAM :

Graphical diagrams it explains what are the roles of each user.



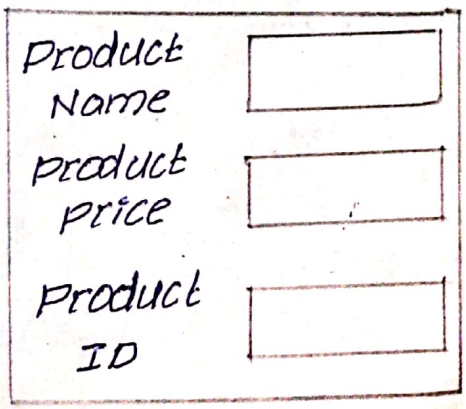
Scanned with CamScanner

3. DATABASE ARCHITECHTURE :



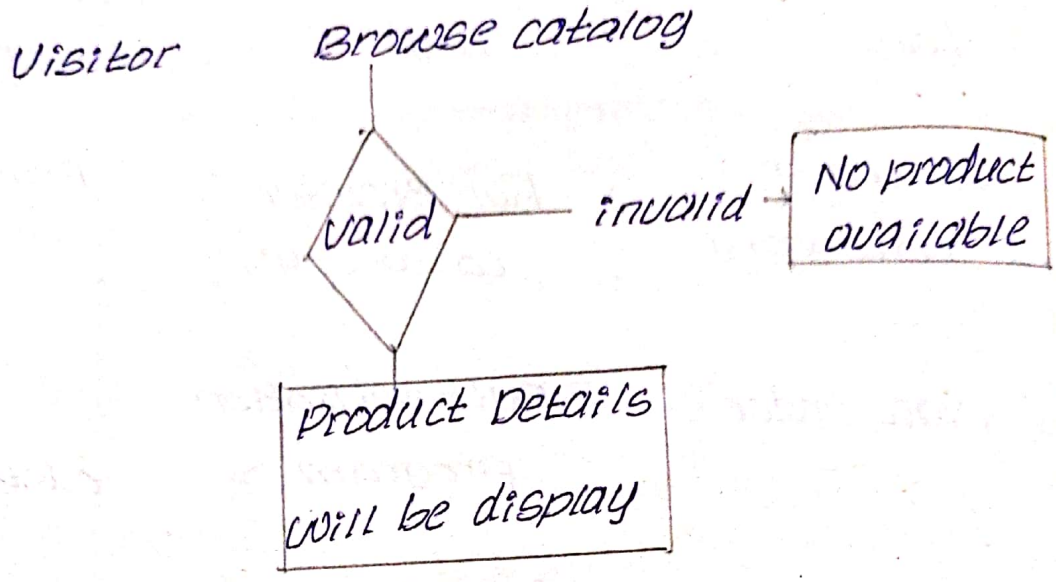
4. DATABASE DESIGN :

Add Product



Scanned with CamScanner

Low Level Design Document:



CODING:

After completed designing and their reviews. developer start programming using Low Level Design Documents (Diagrams).

step① : Database Development

step② : UI Development

step③ : Core Development

Scanned with CamScanner

TESTING :-

After completed program writing development team and testing team test software using below methodologies

1. Whitebox testing
2. Blackbox testing
3. Greybox testing

1. WHITEBOX TESTING TECHNIQUE :-

Whitebox testing done by the deve-lopment team. During this testing development team verify internal structure of source by executing progra-ms.

a. Unit testing
b. Integration testing

2. BLACKBOX TESTING TECHNIQUE :-

Blackbox testing done by the QA team (testing team) during this test tester operate slw functionally without involving in program execution.

a. System testing
b. User acceptance testing

Scanned with CamScanner

[WHITEBOX TESTING] Development team

↓

Testing team

[BLACKBOX TESTING] { QA, QC }

↓

Release

QA → Quality Assurance

QC → Quality control

QUALITY ASSURANCE :-

* Here tester monitor and measure the software development.

* Report any defect to development team any mismatch found.

QUALITY CONTROL :

* Here tester check testing team leakages.

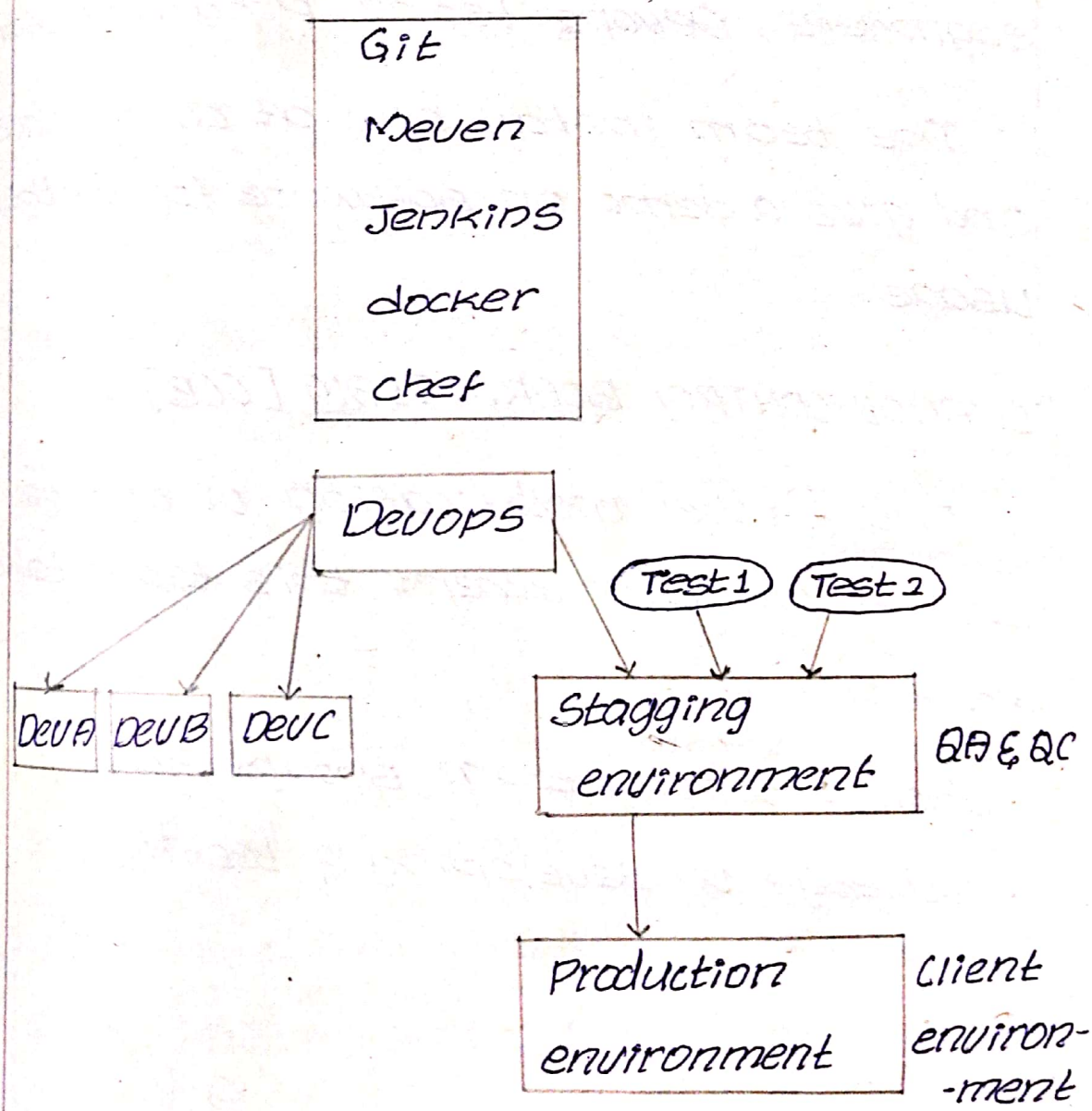
* Development team and Business Analyst cross check whether testing team followed all bylaws to test software or testing team missed any real defects.

Scanned with CamScanner

DEVOPS :-

Development | Operationals

Devops team monitor every developer and tester work day to day and organi-ze continuous integration automation on daily jobs.



Scanned with CamScanner

RELEASE & MAINTAINANCE :-

After completed customer acceptance test organization plan for release software from stagging environment to production environment.

For this release organization form a team. This team includes Hardware engineers, Devops team, Program team.

The team install SW at client place and give a demo on software for further usage.

CHANGE CONTROL BOARD TEAM [CCB] :-

Any further modification or change request on requirement this team take decision.

Change control team should have knowledge on developing & testing.

Scanned with CamScanner

SOFTWARE DEVELOPMENT LIFE CYCLE MODELS

SDLC :-

* Waterfall model
* Iterative model
* Spiral model
* Prototype model
* Rapid development model
* * V model
* * Agile model

WATERFALL MODEL :-

Other name of waterfall model is Linear sequential model. During this model organization develop software in linear fashion.

Waterfall model is older model in software development

In waterfall model all phases accomplished with SDLC phases.

Scanned with CamScanner

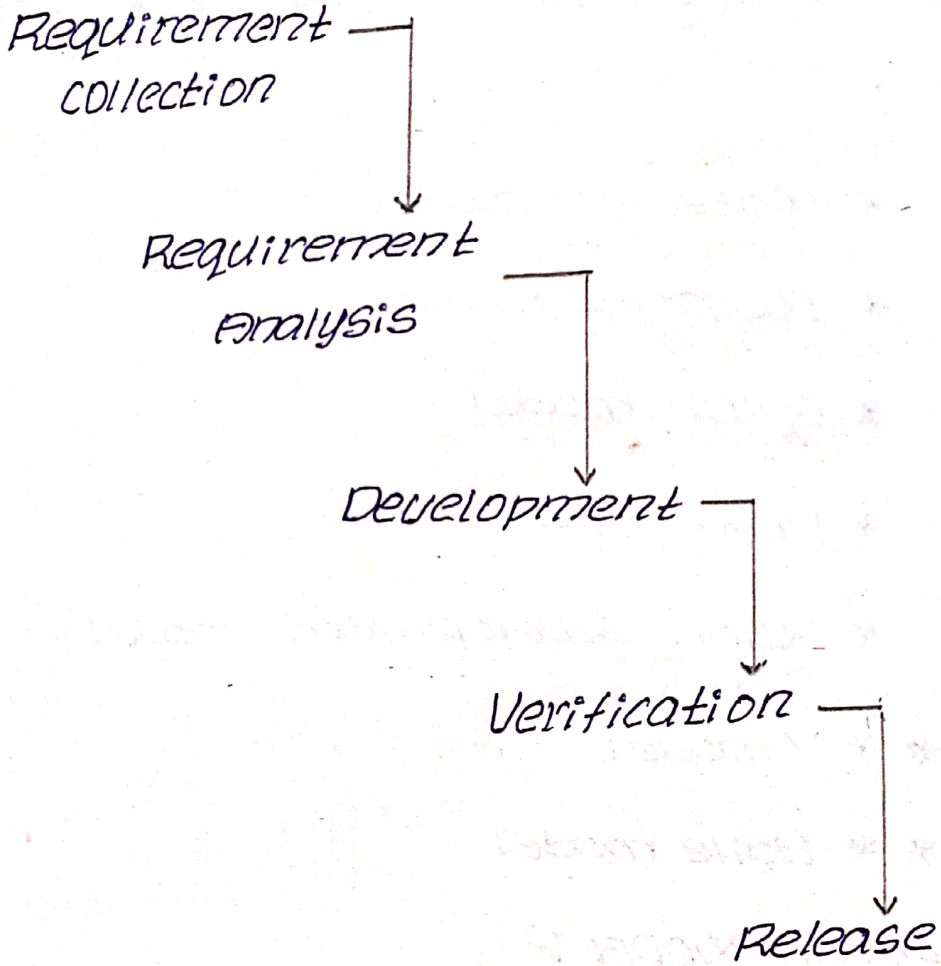


Fig. Waterfall model

ADVANTAGES :-

* Easy to handle smaller project.
* Useful to build known technologies & domain.
* Low risk model.

DISADVANTAGES :-

* Testing team need to wait until all development activities complete.

Scanned with CamScanner

* Untill finish previous phases we can't step into next phase.
* Client can't ask any changes during middle of development.
* To make any changes organization traversing all phase all over again.

ITERATIVE MODEL :

* Also called as multiwaterfall life cycle model.
* During this model software developed in incremental basis.
* Each release is called one iteration
* Based on previous iteration feedback organization start new iteration.

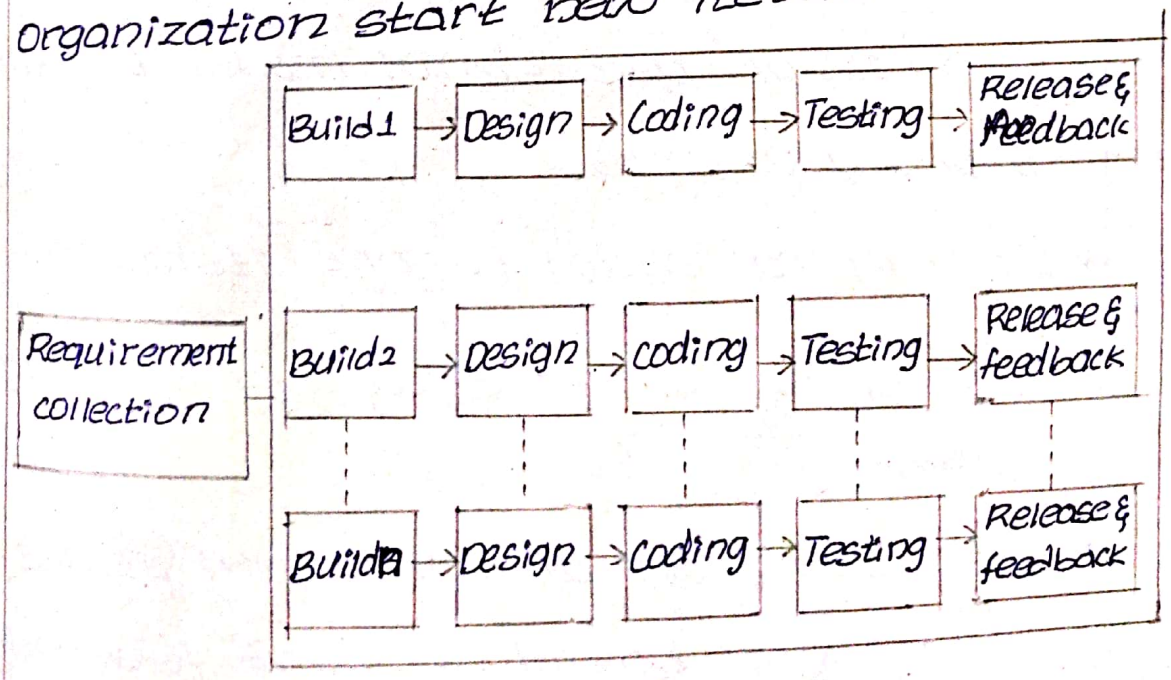


Fig. Iterative Model

Scanned with CamScanner

ADVANTAGES :-

* Each iteration is a short life cycle in SDLC. Testing team involve Quickly after completed development.

* Client can ask any changes after each iteration.

* Smaller iterations are easy to monitor & test.

DISADVANTAGES :-

* Client keep on asking for changes and organization can't afford it.

SPIRAL MODEL :-

* It is a combination of waterfall & Iterative model.

* Organization use spiral model to build mitionary projects.

* Each spiral equal to one iteration.

* Each spiral is one successful delivery to client.

* After every spiral client can utilize software and based on client feedback

Scanned with CamScanner

and market requirement organization plan for next spiral.

* Each spiral passes through planning, risk analysis, development, testing, User feedback.

* Team required highly expertise risk analysis inorder to calculate future risk.

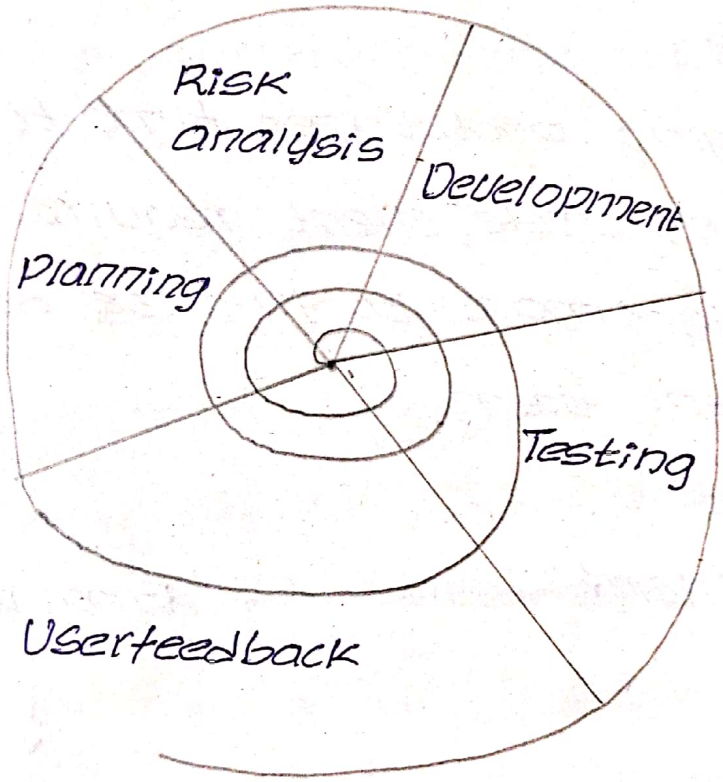


Fig. Spiral model

DISADVANTAGE :-

Too risky model, costly model to use.

ADVANTAGE :-

Short life cycle, useful to build mitionary projects.

Scanned with CamScanner

PROTOTYPE MODEL :-

* Organization use prototype model when client is a laymen, even doesn't know about software development.

* During this time organization built dummy project / prototypes based on client initial requirements. These proto-types helps client to ask real require-ments.

* Changing prototypes time to time based on cirete client requirements. it is a nightmare to finalize design and document them.

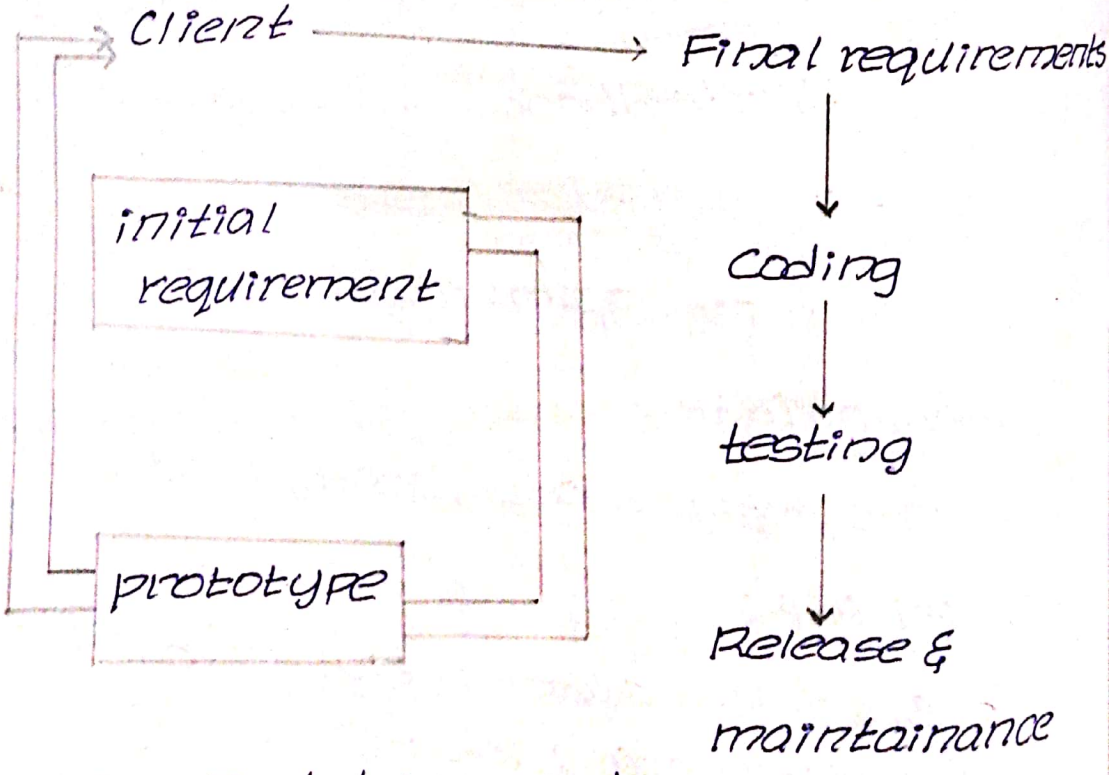


Fig - Prototype model

Scanned with CamScanner

ADVANTAGES :-

* Pictorial diagrams helps client to understand requirements.
* Useful to build new technologies.
* All requirements finalize before development starts.

DISADVANTAGES :-

* If client requirements are keep changing it difficult. maintain proper documentations & design.
* Previously started models as poor record of ending project.

RAPID DEVELOPMENT MODEL :-

* When organization want to develop total software with in the short period of time they follow RAD model.
* Here organization adopt more teams and each team finish design, development, testing same time to complete project early.

Scanned with CamScanner

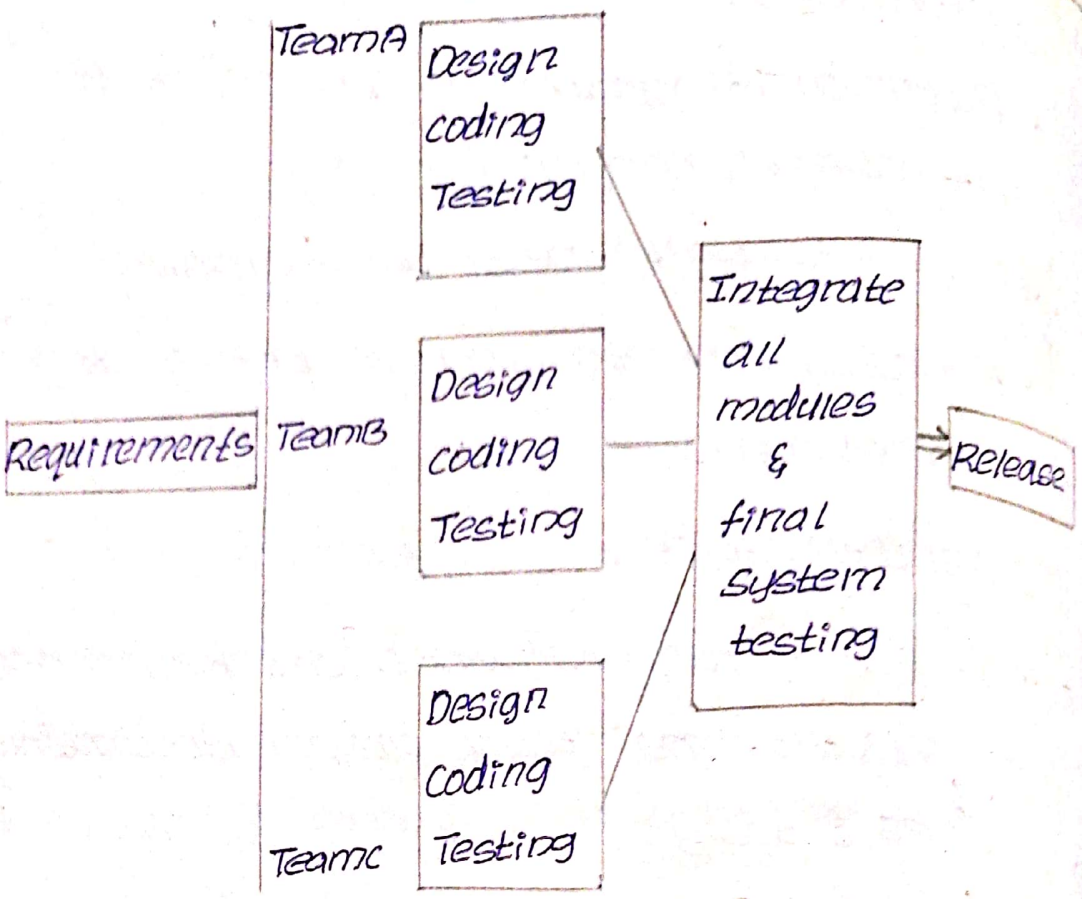


Fig - Rapid Development Model (60-90 days)

V-MODEL :

V stands for Verification and Validation

* During 'V' model testing team has equal weight with development team. It means testing starts early compared to development team.

* In V-model all left side phases incorporated with right side phases

* Coding is bottle neck stage in V-model. Before coding testing engineer involve

Scanned with CamScanner

in static testing - After coding testing engineer involve in dynamic testing

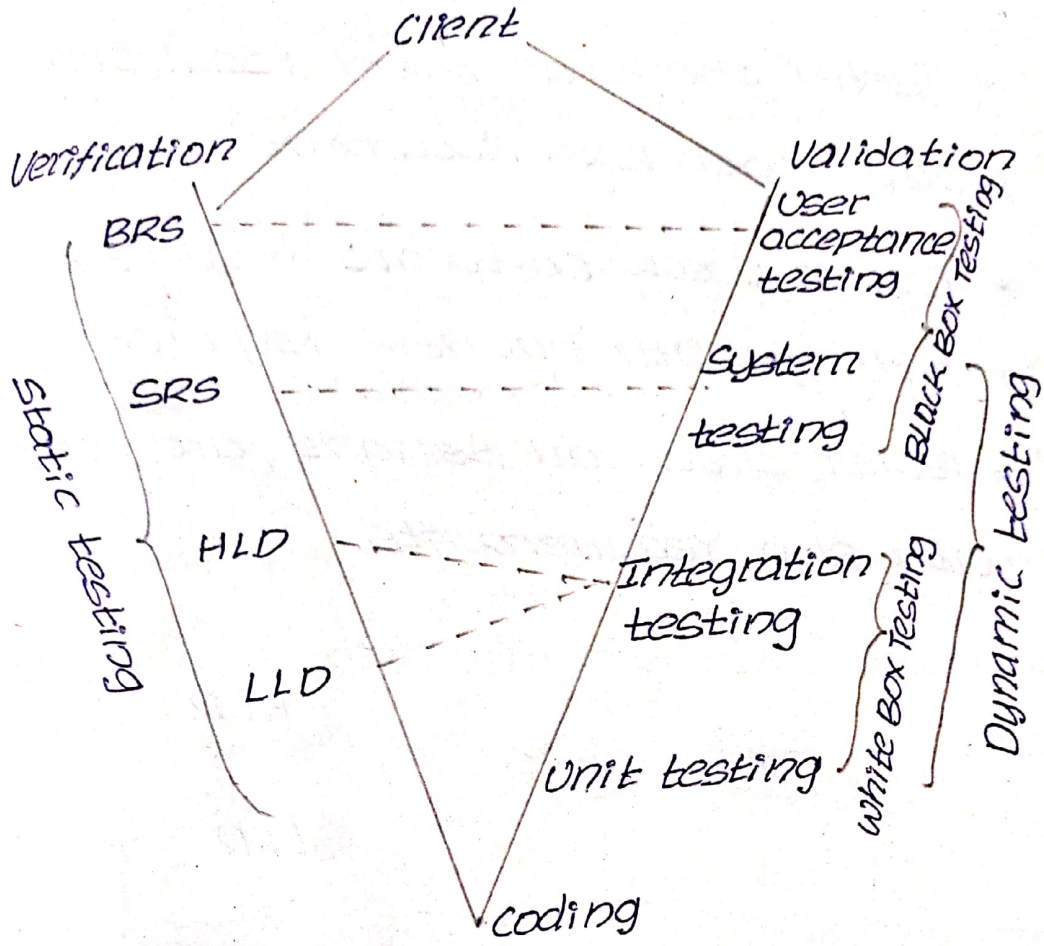


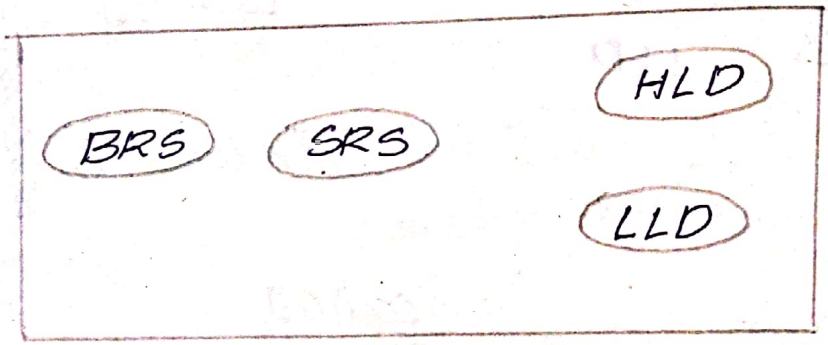
Fig. V-MODEL

Static testing	Dynamic testing
* Document level testing	* Testing conducted on real software.
* Tester verify one we developing right software.	* Tester validate real software to check developed software meeting with client expectation.
*Static testing is a defect preventive approach	*Dynamic testing is a defect detective approach.

Scanned with CamScanner

WHERE TEST ENGINEER INVOLVE IN STATIC TESTING ?

* Tester check all client requirements placed under BRS document
* Tester check software requirements matching with business requirements
* Tester check all designs are matching with SRS requirements.



* Document spell mistakes
* Document index number
* Check tracebility requirements & diagram.

Scanned with CamScanner

AGILE MODEL :-

Agile is a iterative development model. Here iteration is a small period of time upto 1 week to 4 weeks time.

SPRINT :-

Sprint is a duration of time period where selected requirements can develop-ed & tested & release to client successfully.

* Agile is a rapid development model it helps to build software within ~~software~~ short period of time.
* Agile name suggested from agility and adoptability from development.
* Agile is iteration based and each iteration passes through all phases of Planning, design, development, testing.
* During agile project design kept in stone and open at last minute before release software to client.

Scanned with CamScanner

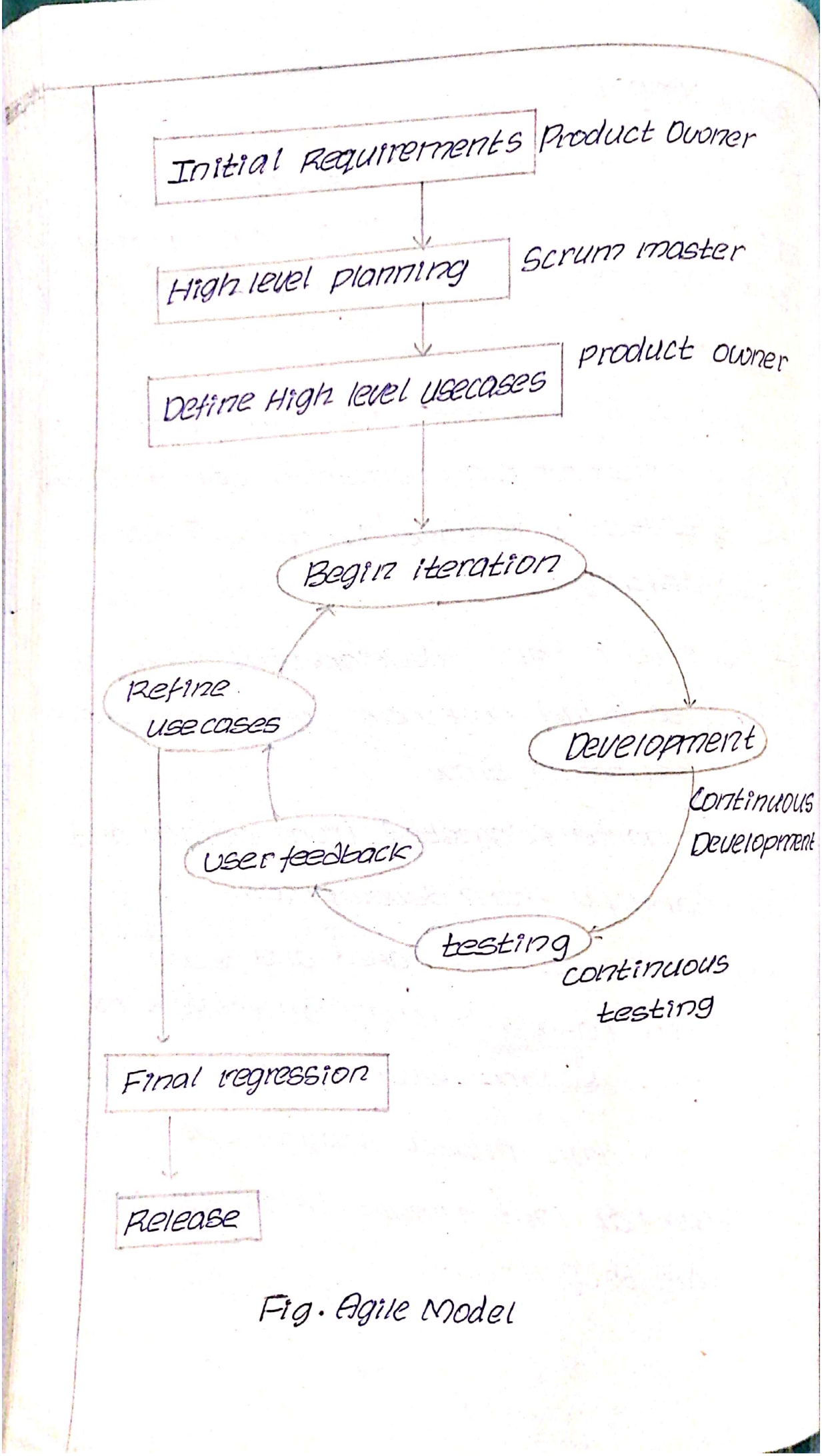
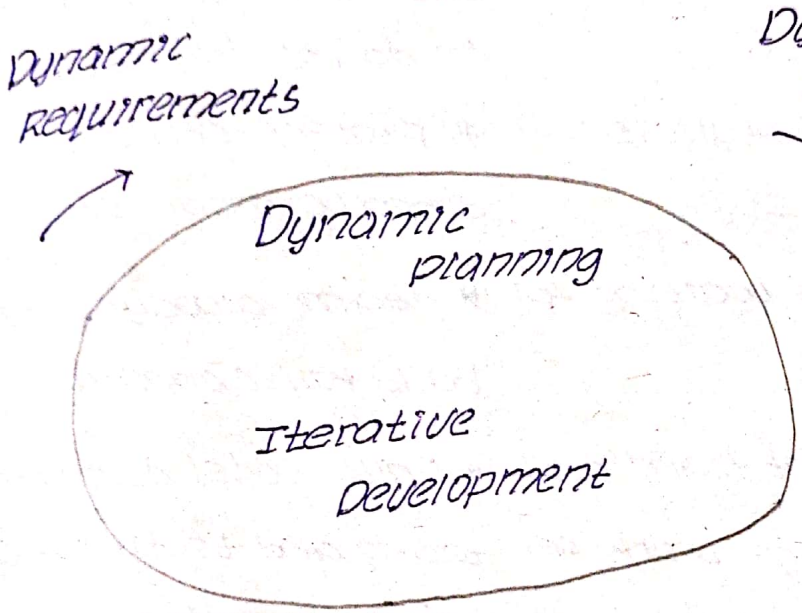


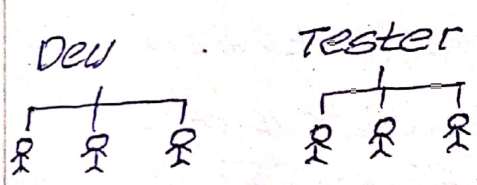
Fig. Agile Model

Scanned with CamScanner

* Because requirements are keep changing agile team follow iterative development plan. [Adhoc plan].



Team Adoption



Continuous development & testing

TESTING IN AGILE MODEL :

In agile model testing also conduct by all team members.

1. Tester
2. Developer
3. Product manager

Scanned with CamScanner

Waterfall model	Agile model
* Linear sequential model. It means development happen phase by phase.	* Iterative development model. It means development happen bits & pieces.
* Client is a Remote user only concentrate on final product	* Client always engages with team and participate in day to day reviews.
* Developing style is controlling style	* Protect style is empowerment style.
* Teams are waiting for deadlines	* Team always try to build killer application.
* Development team and testing team work as batches	* Only model development team and testing team work together.
* Focus on manual testing (Automation is optional)	* Focus on automation testing. test engineer should have at least knowledge of code running
* Mostly junior developers	* Only skilled developers.

ROLE IN AGILE :-

* Product owner
* Scrum master
* Developement team.

Scanned with CamScanner

TESTING METHODOLOGIES :-

1. White box testing
2. Black box testing
3. Glassbox / Greybox testing

1. WHITE BOX TESTING :-

* Developers involve in whitebox testing.
* During this developer verify internal structure of code by executing programs.

    1. Unit testing
    2. Integration testing

2. BLACK BOX TESTING :-

* Testers involve in Blackbox testing.
* During this test testing team validate software functionality over UI-User Interface.

    3. System testing
    4. User acceptance testing (UAT)

3. GLASS BOX TESTING :-

* Gloss box testing is also known as Grey box testing.
* During this test SDET [Software development engineering testing]

Scanned with CamScanner

Engineer involve in test. There Engineer will have knowledge of both development and testing.

* Greybox testers verify S/w by executing programs also operating S/w functionality over UI.

Infosys ⇒ SDET

1. UNIT TESTING :-

* After completed program writing, developer involved in whitebox testing.
* During this testing developers select every individual program and test program using below technologies.

a. Path testing

b. Coverage testing

c. Program technique testing

d. Mutation testing

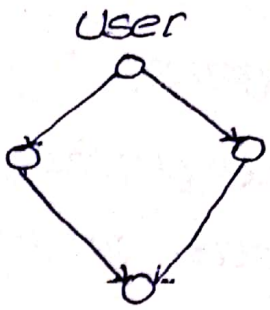
Note :- Before involve in program testing developer design flow graphs.

Scanned with CamScanner

NODES 0

EDGES →

```
Age=22
If(Age>18)
  Accept
else
  Reject
```



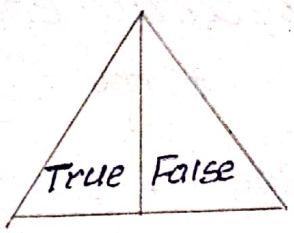
CC = 4-4+2 = 2

CYCLOMETRIC COMPLEXITY :-

E-n+2

a. PATH TESTING :-

* During this test programmer execute each program more than one time to cover all executable areas.



* Other name of path testing is branch testing.

b. COVERAGE TESTING :-

* During this test programmer verify each input and output of program under every branch.

Scanned with CamScanner

* Coverage testing is also known as debugging.

Debugging :-

Everytime of program executing without any syntax errors.

C. PROGRAM TESTING TECHNIQUE :-

* During this test programmer verify execution speed of the program. In case execution speed is not reasonable, developer change internal structure of code without disturb main architecture to improve execution speed of program.

EXAMPLE :-

Execution time = program end time - Program start time

If (Execution time = reasonable time)

FAST

else

SLOW

HOW PROGRAM SPEED IMPROVED :-

* By optimizing code into less lines.
* By removing unwanted conditions.
* Use how memory variable

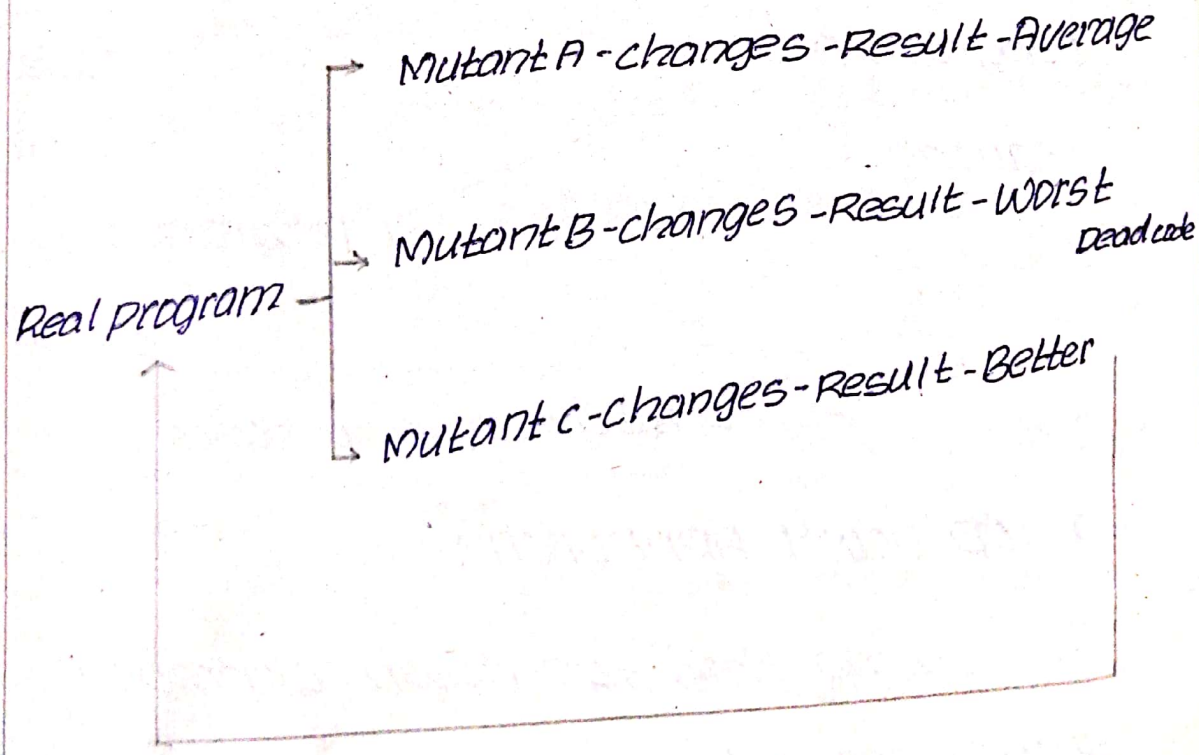
Scanned with CamScanner

d. MUTATION TESTING :-

* During this test programmer create number of Mutant builds. Then make changes on each mutant build and check output matching with the real build.

* If any mutant build performing better than real build programmer replace real build with mutant build.

* If any mutant build performing worst then real build program declare build as dead code and place in trashbin.



Scanned with CamScanner

2. INTEGRATION TESTING :-

After completed unit testing and their code reviews, programmer integrate every individual program with related programs and test each program able to call and connect with related programs

a. Top down approach

b. Bottom up approach

c. Hybrid approach

d. System approach

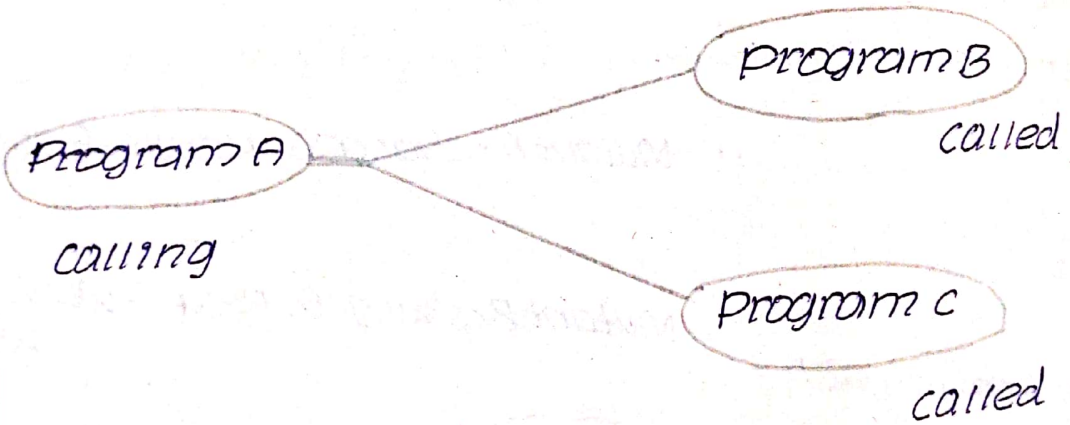


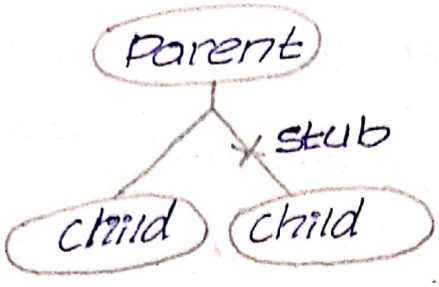
Fig. Integration process

a. TOP DOWN APPROACH :-

During this top-down approach programmer first develop parent programs and then second develop child programs. After they integrate parent and child

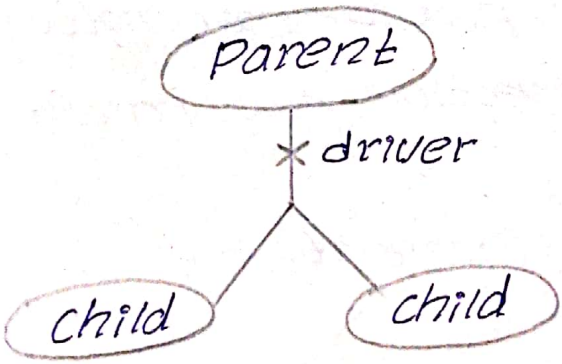
Scanned with CamScanner

programs. Incase any child program still under construction developer use temporary program called stub.



b. BOTTOM-UP APPROACH :-

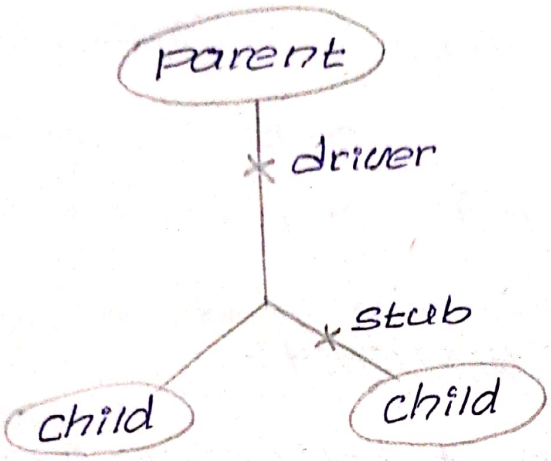
During this bottom-up approach programmer first develop child programs and then second develop parent programs after they integrate parent and child programs. Incase any child prog parent program under construction developer use temporary program called driver.



Scanned with CamScanner

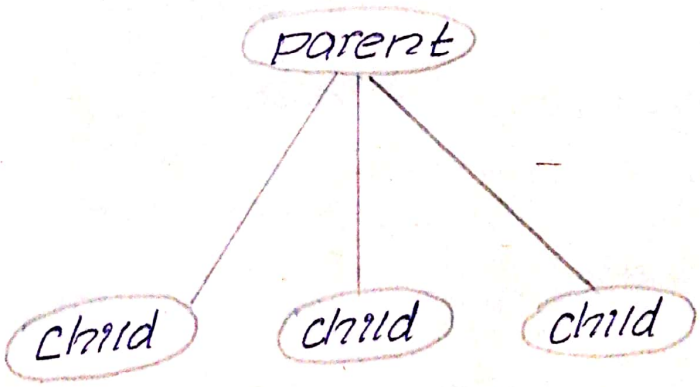
c. HYBRID APPROACH :-

Other name of Hybrid approach is Sandwitch approach.



d. SYSTEM APPROACH :-

* System approach is also known as big bang approach.
* System approach is only possible when all parent and child programs are success-fully developed.
* After successful development of parend and child program, programmer integrate them and test connection blw parent & child programs.



Scanned with CamScanner

SYSTEM TESTING :-

* System testing comes under Black box testing.
* During this test test engineer operate software functionality over User Interface to validate weather software is developed as per client requirement.
* System testing classified into two types.
    * * Functional testing
    * * Non Functional testing

FUNCTIONAL TESTING :-

* Behavioural testing / Graphical User Interface
* Input testing
* Output testing
* Error handling
* DB - testing
* Link coverage testing

NON - FUNCTIONAL TESTING :-

* Usability testing
* Compatibility testing

Scanned with CamScanner

* Performance testing
    - Load / Volume testing
    - stress testing
    - spike testing
    - endurence testing
* security testing
* Inter system testing
* Data Volume testing
* Installation testing
* Globalization testing
* Localization testing
* Multilanguage testing

EXECUTION LEVEL :

- Smoke testing
- sanity testing
- Retesting
- Regression testing
- End-End testing
- Exploratory testing
- adhoc testing

Scanned with CamScanner

FUNCTIONAL TESTING :-

* Functional testing conduct only on GUI (Graphical User Interface).

* Functional area is context based, based on client requirement developer build slw for end user to operate.

* Where end user involve to use software that was functional area.

BEHAVIOURAL TESTING :-

* During this testing test Engineer Verify availability of each screen and each screen response to end user.

* Graphical user Interface testing (GUI) :-

During this testing test Engineer Verify look and feel of software from front page

→ Verify font sizes.

→ Verify colour, background colour of elemen

→ Verify page alignments.

→ Verify image at page.

→ Tables, Forms, Frames, text ... etc

Scanned with CamScanner

INPUT DOMAIN TESTING :

During this testing test Engineer verify whethere software is accepting valid input from end-user.

→ Verify ~~User~~ can enter text into editbox.

→ Verify user can select input from dropdown.

→ Verify user can select radio buttons.

→ Verify user can select checkboxes.

→ Verify user can use mouse and keyboard inputs at software.

→ Verify user can select option from listbox

→ Verify user can select date from date picker.

→ Verify user can select multiple options from list boxes (optional).

→ Verify user able to click buttons, links, selection items.

→ Verify user can operate using touch actions.

→ Verify user able to access sub categories by bovering on main categories.

Scanned with CamScanner

OUTPUT DOMAIN TESTING :-

During this testing Test Engineer operate software using valid credentials / input to verify correctness of output.

→ Typed characters visible at editbox
→ All radio buttons and checkboxes generating rules as expected
→ Verify selected dropdown option populated
→ selected data visible at calender.
→ Verify status messages displayed at webpage after operated software with valid data.
→ Verify expected image / object / text available at webpage.
→ Verify object visible and hidden behaviour according to input entry.
→ Verify checkbox and radio button status after selection and deselection.
→ Verify object enable and disable behaviour according to input entry.
→ Verify expected popup messages displayed to convey status messages
→ Verify expected data records are showing at webtables
→ Verify child feature displayed using after parent entry successful.

Scanned with CamScanner

ERROR HANDLING :-

During this test Test Engineer operate software with invalid inputs to verify software preventing those actions and displayed appropriate error message to end user.

→ Verify appropriate error message displayed on invalid data.

→ Verify popup message displayed on incorrect operation.

→ Verify info message displayed when order is missing.

→ Verify error message displayed when Field leave empty but it expected as mandatory field.

→ Verify table records show "Records not available" when user try empty search or by missing required records.

→ Verify input entry at editbox when text box is in readable mode.

→ Verify user actions on disable elements

→ Verify click functionality of button link when object is inactive.

Scanned with CamScanner

ILLEGAL HANDLING :

During this test Test Engineer purposly hack software with illegal approach and whether software is preventing user to access.

→ To many invalid attempts to access password.

→ Guessing passwords.

→ Verify page not allowed user to access page source.

→ Verify user able to take screenshot of page when it is not allowed.

DB TESTING :

Also known as Backend testing.

DB testing mainly categorized into two types.

1. Data mapping
2. Data integration

1. DATA MAPPING :

* Data Mapping is also known as data validity.
* Checking connection of new database tables and forms.
* Verify that respective tables and records are updated when user clicks 'save', 'update', 'search', or 'delete'.

Scanned with CamScanner

2. DATA INTEGRITY :-

Consider that different modules (i.e. screens or forms) of application use the same data. In that case, make it sure that the latest state of data reflected everywhere.

NOTE :- System integrity or End to End system integration testing done by functional testing team at user interface of software.

COMMON TEST CASE FOR DB TESTING :-

→ Check if all the triggers are in place to update reference table records.

→ Check if any incorrect / invalid data exists in the major columns of each table.

→ Try to insert wrong data in tables and observe if any failure occurs.

→ Check what happen if you try to insert a child before inserting its parent.

→ Check all passwords are encrypted type and should not allow user to view password at master database.

→ Check insertion of duplicate data when primary key active.

Scanned with CamScanner

LINKS COVERAGE :-

Links play keyrole in webapplications to test links at webpage fellow below checklist.

→ Verify if any broken images and links at webpage.

→ Verify administration links operating thirdparty tools.

→ Verify all external link open page at private window.

→ Verify all hyperlinks connect intersystem softwares.

→ Verify secure pages opening in incognito window.

Scanned with CamScanner

NON-FUNCTIONAL TESTING :

* Tester involve in non-functional testing using NFRS document [Non functional requir-ement Specification].

* After completed functional testing tester involve in non-functional testings. These non-functional are support functional to perform better.

USABILITY TESTING :

During this testing Test Engineer verify user friendlyness of software.

NOTE : Test Engineer verify software user friendlyness using below checklist.

→ Verify Spelling in all screens of software.

→ Verify labels in it all screens of software.

→ Verify meaning of labels in all screens of software.

→ Verify line spacing in Label and control.

→ Verify line spacing in controls.

→ Verify location of controls which are functionality related.

→ Verify OK and cancel button in every screen

Scanned with CamScanner

→ Verify shortcuts in labels of screens.

→ Verify icon tool tips.

→ Verify keyboard access on every screens.

→ Verify meaning of error

→ Verify system menu existance in screens.
(Minimize, Maximize, Restore, click ... etc).

→ Verify help documentations.

→ Verify color contrast of screens.

COMPATABILITY TESTING :-

During this test tester verify browser and operating system compatability.

Here Test Engineer verify whether software is accessable and operatable with in client expected browsers and operating systems.

NOTE :-

Other name of compatability testing is cross browser testing.

[In real time organizations use selenium kind of automation tools to conduct cross browser automation].

Scanned with CamScanner

PERFORMANCE TESTING :-

* Performance testing is non functional testing.
* performance testing only possible using automation tool like [Load Runner, Jmeter]
* During this test performance testing engineers verify software speed and response under customer expected configuration.

performance testing can be conducted using below sub tests

1. Load testing
2. Stress testing
3. Spike testing
4. Endurance testing

LOAD TESTING :-

The execution of software under client configuration and client expected load (No of concurrent users) to estimate speed in processing is called load testing

STRESS TESTING :-

The execution of software under client

Scanned with CamScanner

configuration and more than client expected load to estimate "peak load" is called stress testing.

SPIKE TESTING :-

The execution of software under client and more than client expected load to estimate reliability.

During this test tester add no of current user suddenly and verify speed in processing

ENDURANCE TESTING :-

The execution of software under customer configuration and more than customer expected load continously to estimate logvity and durability is called endurance testing.

DATAVOLUME TESTING :-

During this test, testing team is operating software by storing sample data to estimate capacity of the software database.

Depends on technology. database capacities are changing due to this reason testing team conduct data volume testing to find the capacity of database in number of records.

Scanned with CamScanner

INTERSYSTEM TESTING :-

During this test, testing team verify our software is sharing data and co-existing with other softwares.

INSTALLATION TESTING :-

* Installation testing conduct most commonly on Desktop softwares.
* To conduct installation testing tester follow below steps.

→ Verify set up program execute to start installation.

→ Verify friendliness of screens while installation

→ Verify occupied space after installation

→ Uninstall

→ Verify user can cancel installation any time.

→ Verify installation asking accept license agreement.

RECOVERY TESTING :-

Recovery testing is performed in order to determine how quickly the system can recover after the system crash or hardware failure. It comes under the type of non-functional testing.

Scanned with CamScanner

GLOBALIZATION TESTING :

Globalization is a process of adapting globalization software for a specific region or language by adding local specific components.

→ verify software available in local language.

→ verify local timezones are displaying.

→ verify local maps displaying at software

Multy languity testing :

Some times developers are developing a software in java unicode or .net with unicode this type of software provide output in different language and by taking input in one language.

GLOBALIZATION TESTING :

Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.

Scanned with CamScanner

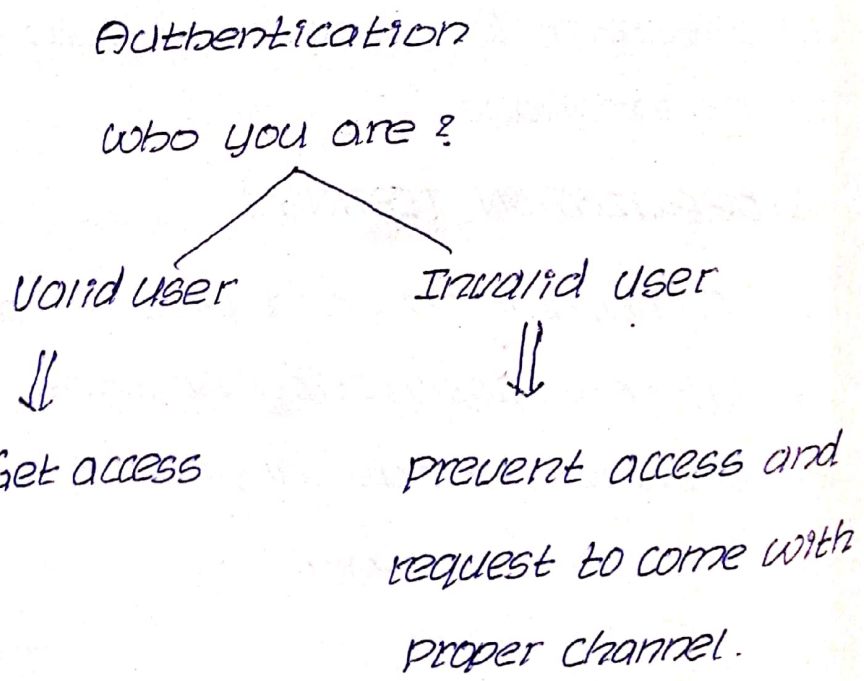
SECURITY TESTING :-

Security testing is a process of finding weak spots of software. Here test engineer verify software flaws and how illegal users hack the software with malicious data.

1. Vulnerability testing
2. penetration testing

AUTHENTICATION :-

Tester verify for valid user system providing access and preventing access to invalid user.



AUTHORIZATION :-

Tester verify according to the role user getting access or not.

Scanned with CamScanner

Authorization

→ Are user getting valid rights to use software.

→ Are software preventing some right which is not related to the user.

COOKIES TESTING :-

While Software using in different browsers. Browsers save history of transaction like remembering passwords, previous transactions and history.

Here test team verify all these browsers deleting previous transaction history automatically when security protection was enable.

→ Tester verify are cookies still maintain history after enabled auto delete option.

→ Are cookies allowed user to delete manually.

→ Are Browser allowed cookies to add manually.

→ Are cookies are expired after expiring data elapsed.

Scanned with CamScanner

BRUCE FORCE ATTACK :-

It is a automation security testing. During this tester try to access authentic. -ation by sending a series of combinations.

SQL INJECTION :-

* SQL injection is also known as data trancation
* Here tester change data by sending malicious code and try to change real data acceptance values.

USER ACCEPTANCE TESTING :

After QA - testing Signoff before releasing software to client organization team and client team involve in testing to control detect leakages from testing team.

1. Alpha testing
2. Beta testing

1. ALPHA TESTING :-

Type ⓐ ⇒ Invite client team to visit organ- -ization and team involve in testing to find leakages from testing team.

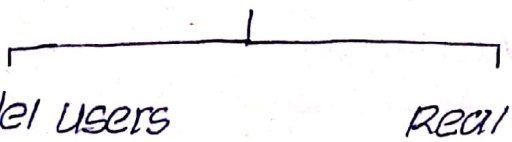
Scanned with CamScanner

Type ⓑ ⇒ Developing team and Business analyst verify are testing team follows all bylaws to test software and are they failed to find hidden deffects in software.

* Alpha testing is a inhouse testing or factory level testing. here developing, testing and Business analyst all together try to findout defect missed by testing team.
* After completed alpha testing organization plan for software release to production environment.

2. BETA TESTING :-

Beta testing conducted in production environment by the client team.



MODEL USERS :-

Here client choose selected people and train them on software. Then model user use software before releasing to market

* Technical experts
* Industry experts

Scanned with CamScanner

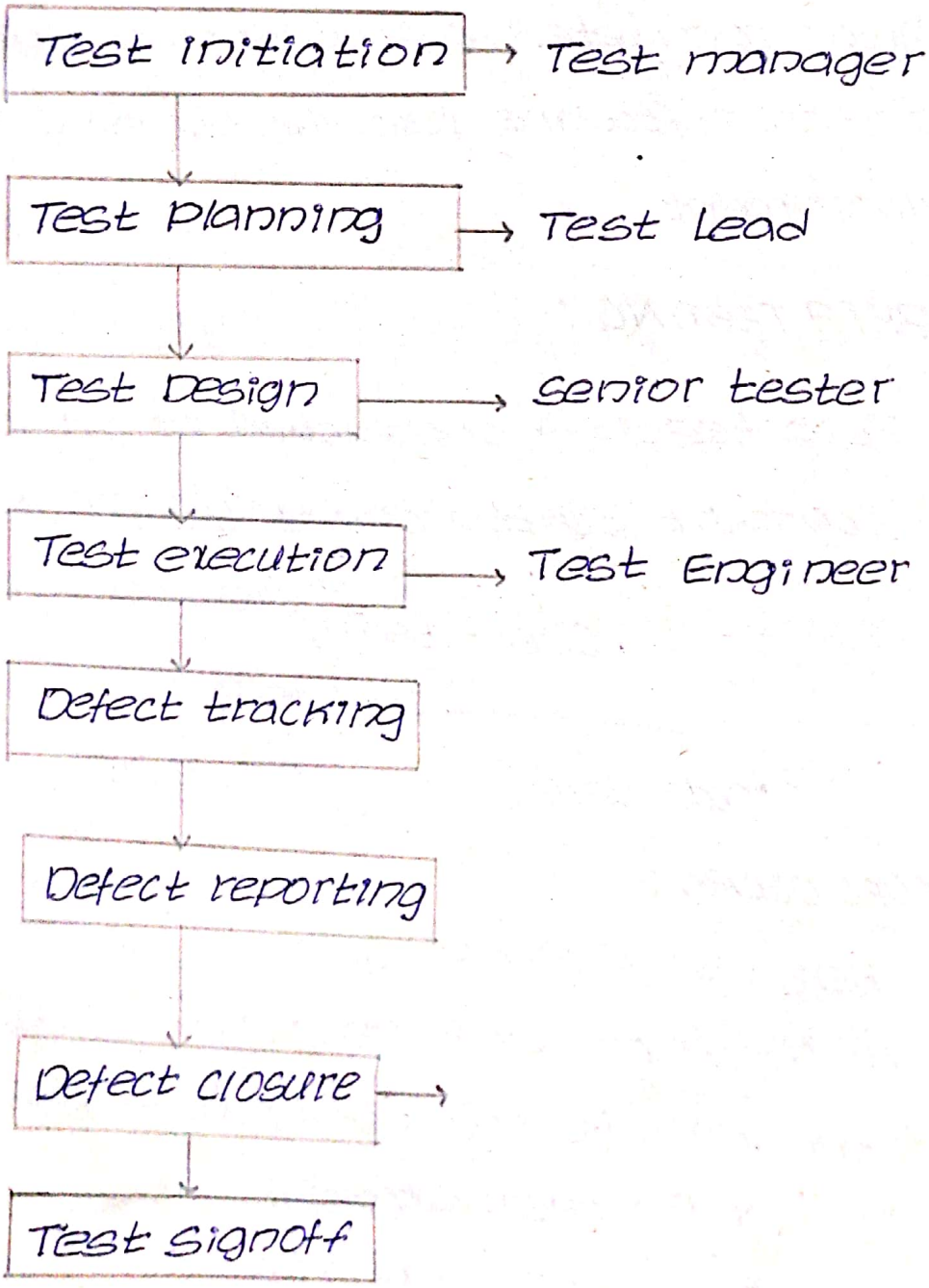
REAL USERS :

Client recruit real public and ask them to user software.

Client release temporary edition of sw to limited user, inorder to find user detecting problems.

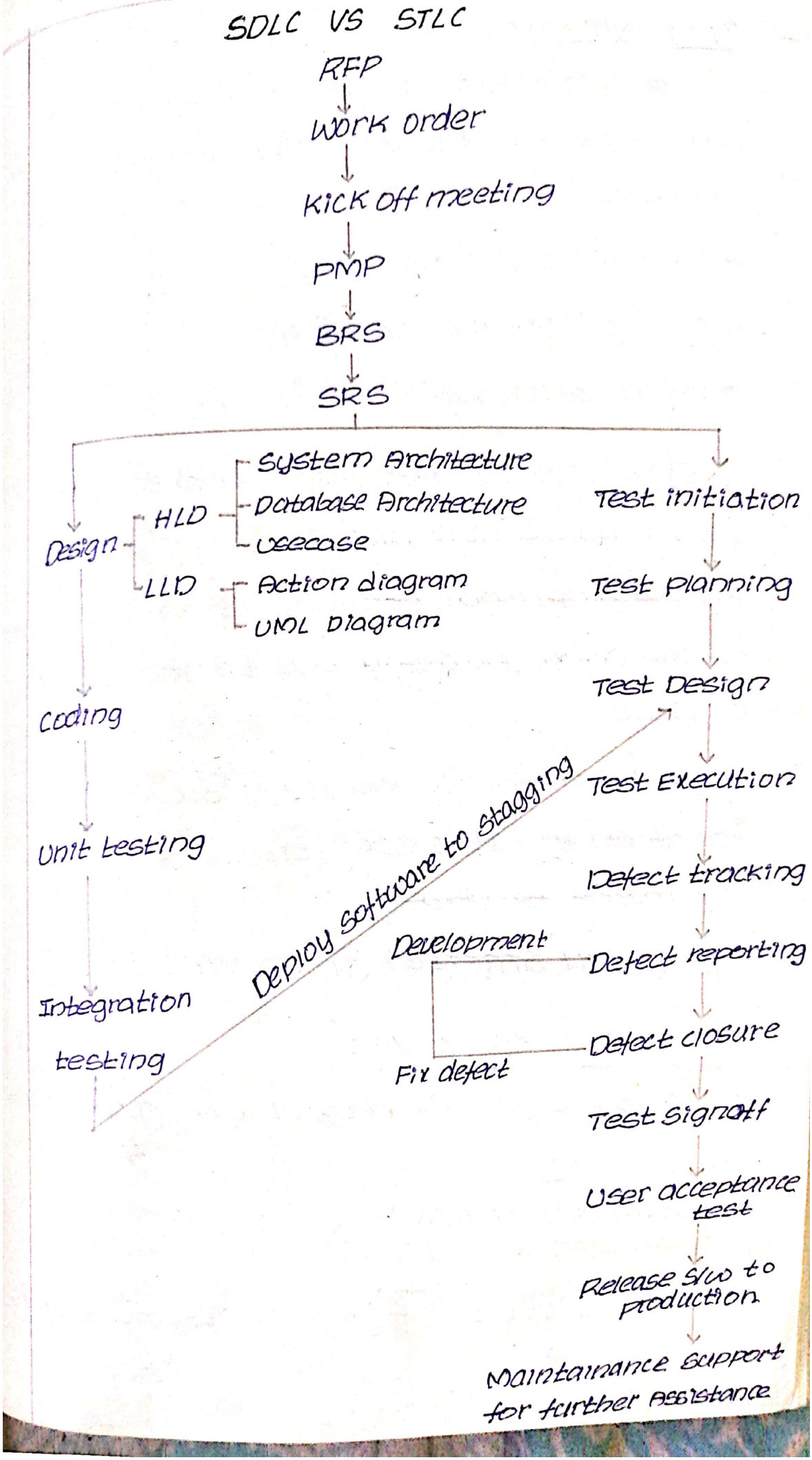
SOFTWARE TESTING LIFE CYCLE (STLC) :-

Systematic approach to conduct system testing from start to end.



Scanned with CamScanner

SDLC VS STLC



Scanned with CamScanner

TEST INITIATION :-

It is a initial phase of STLC, Here test manager, project manager prepare strategies.

* Exhastive strategy
* planned optimal strategy
* Adhoc strategy

Exhastive strategy is impractical so that organization always follow adhoc planned optimal strategy. Some organizatio follow adhoc strategy due to inavailability of time.

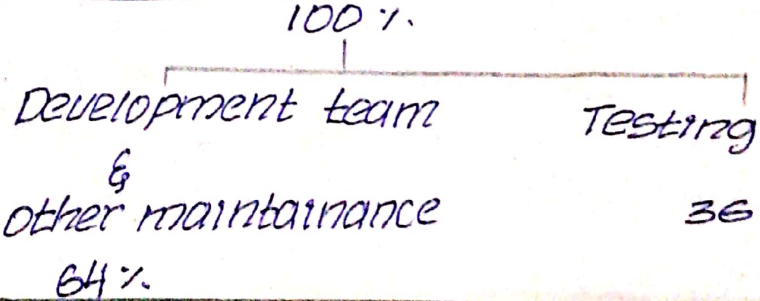
QA team follow strategy document prep-aration in IEEE 829 format. 829 is for document templates.

ITEMS IN STRETEGY TEMPLATE :-

SCOPE AND OBJECTIVE :-

Importance of project and importance of testing

BUDGET ALLOCATION :-



Scanned with CamScanner

TEST RESPONSIBILITY MATRIX :-

Name of types of testings under project

* System testing
* performance testing
* API testing
* DB testing

ROLES & RESPONSIBILITIES :-

Names of the jobs and their responsibilities

Test Lead :- Design test plan coordinate with testing team & test manager.

Senior tester :- Design test cases and participate in reviews

* Coordinate with test lead
* Defect reporting

Junior test engineer :-

* Execute testcases
* Defect reporting
* Coordinate with test lead & Sr test engineer

COMMUNICATION TOOLS :-

Required negotiation channels between two tester's / Jobs.

Scanned with CamScanner

AUTOMATION TOOLS REQUIRED :-

Names of required automation tools

* Selenium
* postman

TRAINING PLANS :-

* Training required on tools and duration of time
* Shortlisting people for trainings

CONFIGURATION MANAGEMENT TOOL :-

* Name of tool where we can find project documents like BRS, SRS, HLD, LLD documents.
* Programming code
  - → Visual source safe
  - → cloud
  - → Git
  - → bit bucket

  } Repository and version controlling tools

RISKS AND MITIGATIONS :-

* Possible risks and mitigation plans [how to overcome risk].

TACTICAL RISK :-

* Lack of time
* Lack of testing knowledge on specific tools
* Lack of communication between teams

Scanned with CamScanner

TEST PLAN :-

Test lead design this document and this is a guidline document to testing team

what to test ?

when to test ?

where to test ?

How to test ?

PROJECT OVERVIEW :

Purpose of project, overview of project

TEST PLAN ID :-

Version number of test plan

TP-V1.0

REFERENCE DOCUMENTS :-

BRS, SRS

FEATURES TO BE TESTED :- (SCOPE)

Types of testings and areas Of testing where tester should involve

Types of testing	Area of testing
*Functional testing	* Admin { Create, delete
* API testing	customer

Scanned with CamScanner

FEATURES NOT TO BE TESTED :- (OUTSCOPE)

* Testing is already done { Unit testing, integration testing

* Types of testing and areas not part of your role.

performance, security } Others teams involves in this testing

APPROACH :

A document description of total testing approach from start to end.

ENTRY CRITERIA :-

what are the process to follow before start Software testing

* Unit & Integration should be finished
* Test case should reviewed.
* Software installed into stagging environment.
* Test plan released to testing team

SUSPENSION CRITERIA :

When testing team can hold testing (or) suspend test.

* Suspended test when smoke testing

Scanned with CamScanner

failed.

* Hold execution when previously reported defects are not fixed.
* For any environment trubles.

RESUMTION CRITERIA :-

[When to resume suspended execution]

* After environment problems fixed.
* previously reported problems fixed.
* Software recovery from failure.

EXIT CRITERIA :

[To stop test execution]

* All the requirement in software are tested.
* Test duration exceeded.
* 95 % test cases passed.
* All major defects should be resolved

[Take screenshot of each test pass/fail].

TEST DELIVERABLES :-

The list of testing documents to be prepared by test engineer for future reference [Test scenarios, test cases, test logs, defect reports, Test plan etc).

Scanned with CamScanner

ASSUMTIONS :-

* Stagging server should be available with offshore team.
* All necessary login credentials, accessable for QA team.
* Developer contact details obtain for QA team for urgent fixes.
* Proper training should arrange to testing team on new technologies.

RESOURCE TRAINING :-

The names of selected test engineers for the corresponding project testing and required no. of training sessions for them to understand customers requirements.

RESPONSIBILITIES :-

The work allocation to testing team [according to their technical skill].

SCHEDULES :-

* The date and time to be followed by test engineer
* Future build release dates and shedules.

Scanned with CamScanner

RISKS & MITIGATION :-

* The previously analyzed list of risks and solutions to overcome
* what if time is exceeded ?

  perform informal testing [Adhoc & exploratory testing].

DEFECT TRACKING PROCESS :-

→ State of defects

  open, Reopen, closed

→ what seviourities to follow

  Urgent, fatal, high, medium etc

→ priorities to forlow

  could fix, should fix ... etc

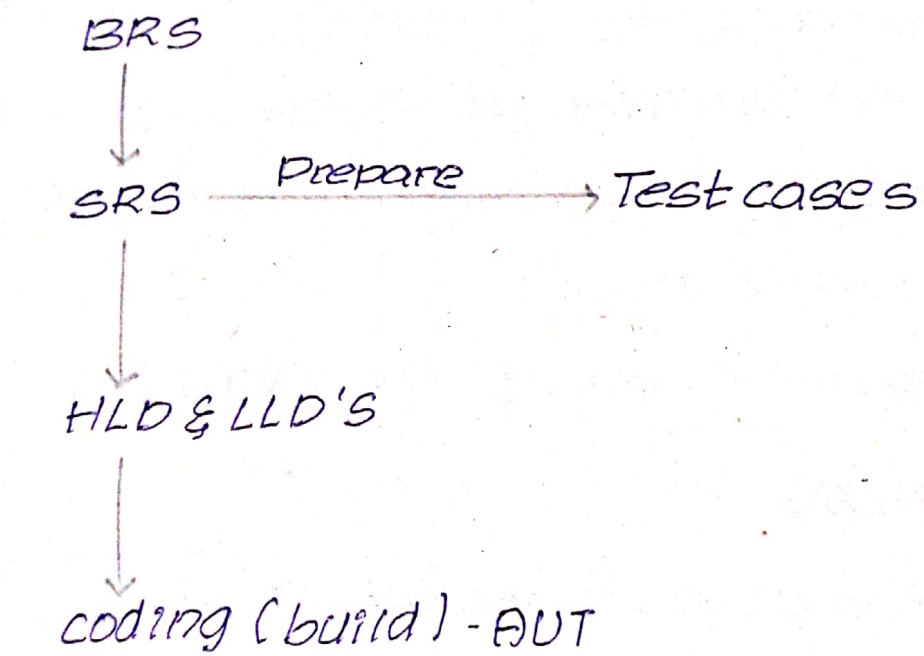
APPROVALS :-

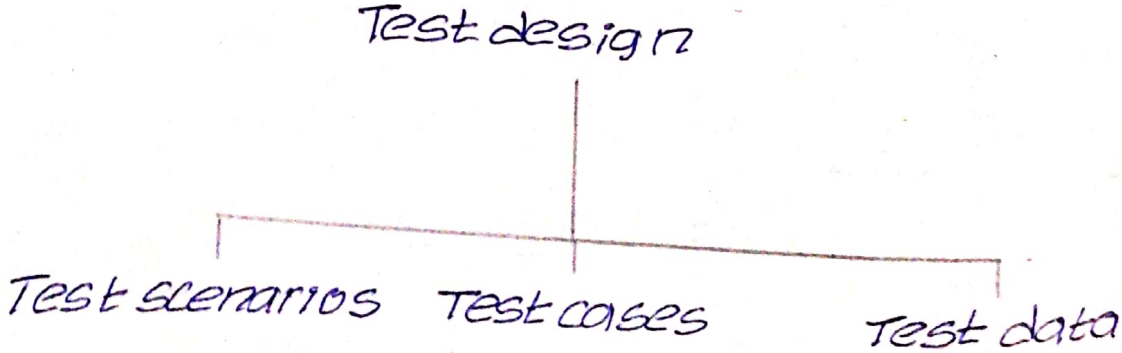
Approvals from project Manager.

Scanned with CamScanner

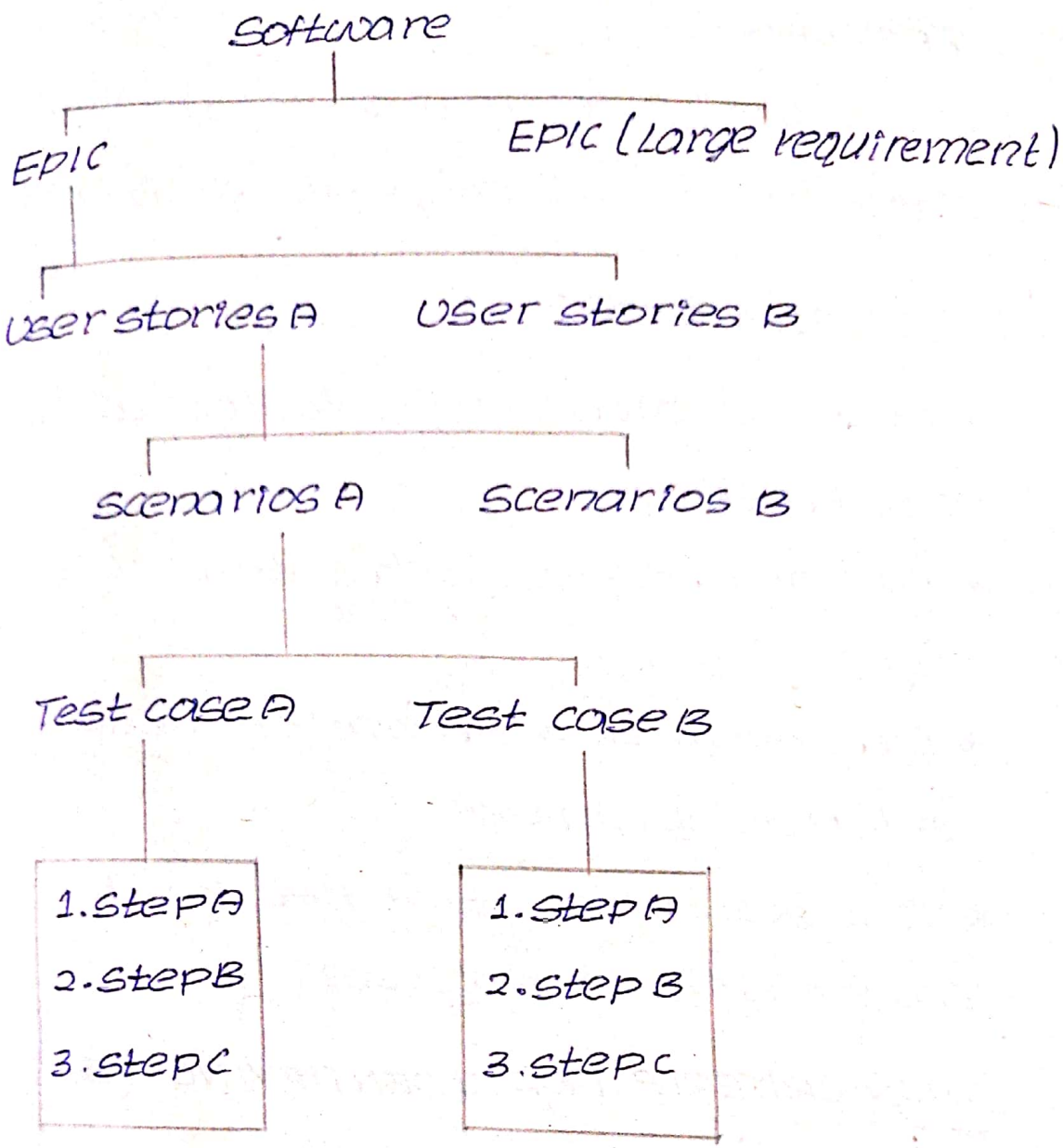
TEST DESIGN :-

After completed SRS design organization forward software requirements to development team and testing team. When development team involve in designs & codings. Testing team identify scenarios to test & design test cases using black box techniques.





Scanned with CamScanner



EPIC :-

Epic is a larger requirements. Each Epic contains set of user stories

USER STORY :-

A tiny part of requirement what client wants

Scenario :-

Feature to be tested

Scanned with CamScanner

TEST CASE :-

A test case is a set of user actions and subsequent response from the system.

TEST SCENARIOS :-

* Possible areas to be tested (or) what is to be tested.
* Scenario defines what to test in user story.
* In general test scenarios listed in test plan document.
* Test scenarios identified by the test lead or Sr. test engineer.

ENTRY CRITERIA FOR IDENTIFYING TEST SCENARIOS :-

* Approved test plan.
* Approved SRS
* Test scenario template
* Any design documents available (If any)
* Any blue prints available (If any).

EXIT CRITERIA FOR TEST SCENARIOS:-

* Test scenarios should reviewed & Approved
* Once test scenario are approved test lead will create a baseline for test scenario (TS1.0

Scanned with CamScanner

and he will update scenarios into common repository.

SCENARIO TEMPLATE :-

Test Scenario ID	Document Reference ID	Description	Scenario title	Child scenario	UI Developer	Scenario Develop-ment / status	Possible count of test cases
TS001-CM	SRS V1.0 10.0 ⇒	Origin of requirement	Search	NA	Yes	No	5
		"	Add to cart	Available Out of stock coming soon	Yes	No	15
TS002-CM	SRS V1.0 10.1 ⇒						

Scanned with CamScanner

TEST CASE :-

* A test case is a set of user actions and subsequent response from the system.

* Every positive and negative action at software what subsequent response receive from system we design as test case document.

* Here tester prepare testcases as checklist when software is under development. once software is ready tester use same checklist to verify is software developed as client expected.

ENTRY CRITERIA TO PREPARE TEST CASE :-

* Approved test plan.

* Approved SRS

* Approved FRS (or) user case template (user case template is ideal way to design testcase).

* Approved test scenarios

* Test case template.

Scanned with CamScanner

EXIT CRITERIA FOR TEST CASES :-

Test cases should be reviewed and approved.

TEST CASE CONTAINS :-

A test case should contains particular such as

* Objective
* Test conditions
* Input data
* Expected results
* Location of the files to be used in test case (Test scenario reference numbers document reference numbers).
* trouble shooting guidelines

GOOD TEST CASE DESIGN :-

A good test case satisfies the following criteria

* Effective - Finds Faults [objective]
* Evaluable - Easy to maintain
* Requirement coverage
* Requirement completeness
* Easy to understand

Scanned with CamScanner

* Should not be out of scope - do not contain unnecessary things.
* Testcases should write in uniform way.
* If you started testcases for one requirement don't merge other requirement.

TEST CASE DESIGN TECHNIQUE :-

WHY TEST TECHNIQUES ?

* Exhaustive testing (use of all possible inputs and conditions) is impractical.
* Need thought processes that help us to select test cases more intelligently.
* Test case design techniques helps to cover more areas to test s/w.

ADVANTAGE OF TECHNIQUES :-

* Different people : similar probability of finding faults.
* Effective testing : To find more deviations (faults).
* Once can focus or pay more attention on specific types of faults.

Scanned with CamScanner

* To know you are testing right thing
* Avoid duplication.
* Identify the best possible combinations to cover maximum conditions.

Test case design techniques are broadly categorized into 2 :-

* Blackbox (functional) test case design technique
* Whitebox (structural) test case design technique
* Experience testing (informal testing).

BLACK BOX TEST CASE DESIGN TECHNIQUE :-

* Equivalance class partitioning
* Boundary value Analysis
* Decision table
* State transition technique
* Error guessing

Scanned with CamScanner

EQUIVALANCE PARTITIONING :-

* Equivalence partitioning is a method for deriving test cases. In this method given input is divided into number of equivalence classes.

* From each equivalence class one input value is choose for testing.

* Equivalent partitioning drastically cuts down the number of test cases.

* We can label the ECP classes as "valid" and "invalid"

EXAMPLE :-

Consider any edit field which can accept values between 4-12.

From the given range of input value we can form 3 equivalence classes

i. less than 4

ii. Between 4-12

iii. Greater than 12

consider edit field accept only Alphabets

* Valid : Alphabets [a-z] [A-Z]

* Invalid : Numericals, Special characters, spaces.

Scanned with CamScanner

* Consider edit field accept only Alphabets and Numericals, At least one numeric value.

* Valid: Alphabets [a-z] + Numerics [1]

* Invalid: Only Numerics, Only Alphabets, Special characters, spaces, Blank

Consider edit fields accept only Alphabets should starts with uppercase and should contains at least one special characters

* Valid: Alphabets starts with uppercase characters, Alphabets with one special characters

* Invalid: Alphabets starts with lowercase. Alphabets without special character, Numbers, spaces, starts with Numbers, starts with special characters.

Scanned with CamScanner

User Name	Role	Rights
Abbay	Financial prof	Assign
Sriram	customer	View, Add, Update
Ravi	customer	View, Add, Update
Dinakar	Financial prof	Assign
Bhaskar	Employee	View, Delete, Add, Update
Ramya	Financial Prof	Assign
Pranathi	Financial prof	Assign
Harsha	Employee	View, Delete, Add, Update
Suchitra	Agent	View

→ Identify similar functionalities and divide equally

→ Divide the inputs, outputs, etc., into areas which are same

→ Here the Assumption is, if one value works then all should works

→ Identify one from each partition instead of all from one.

Role	User Name	Rights
Financial prof	Abbay	Assign
Customer	Sriram	View, Add, Update
Employee	Bhaskar	View, Delete, Add, Update
Agent	Suchitra	View

Scanned with CamScanner

EXAMPLE :

Specifications state that a max of 4 purchase orders can be registered against any one product

The equivalence classes are :

Valid class : 1-4 purchase orders

Invalid classess : # purchase orders > 4,

purchase orders < 1

EXAMPLE :

→ Customer Name - 2 to 64 chars

→ Account Number - 6 digits, first digit is non zero

→ Loan Amount requested - $500 to $9000

→ Loan Term - 1 to 30 years

→ Monthly Repayment - Min of $10

Customer Name

Conditions	Valid partitions	Invalid partitions	Valid Boundaries	Invalid Boundaries
Customer Name	2 to 64 chars Valid chars	< 2 chars > 64 chars Invalid chars	2 chars 64 chars	1 char 65 chars 0 chars

Scanned with CamScanner

ACCOUNT NUMBER :-

Conditions	Valid partitions	Invalid partitions	Valid Boundaries	Invalid Boundaries
Account Number	6 digits 1st non-zero	<6 digits >6 digits 1st digit = 0 Non-digit	100000 999999	5 digits 7 digits 0 digits

LOAN AMOUNT

conditions	Valid partitions	Invalid partitions	Invalid partitio Boundaries	Invalid Boundaries
Loan Amount	500 - 9000	< 500 >9000 0 Non-numeric	500 9000	499 9001

BOUNDARY VALUE ANALYSIS :-

→ In this method tester has to concentrate more on the boundaries of the input values.

→ In testing boundary conditions have a higher probability of detecting errors.

→ Here tester verify the faults at near boundaries

Scanned with CamScanner

EXAMPLE :-

→ Consider any edit field which can accept values between 18-35.

→ According to BVA method the valid inputs are

* 17, 18 .. 35 & 36

→ If the range is 'a' to 'b' then valid inputs are

* a-1, a, a+1, b-1, b & b+1

EXAMPLE :-

consider zipcode editbox accept Max number 6 only.

According to BVA testable boundaries are

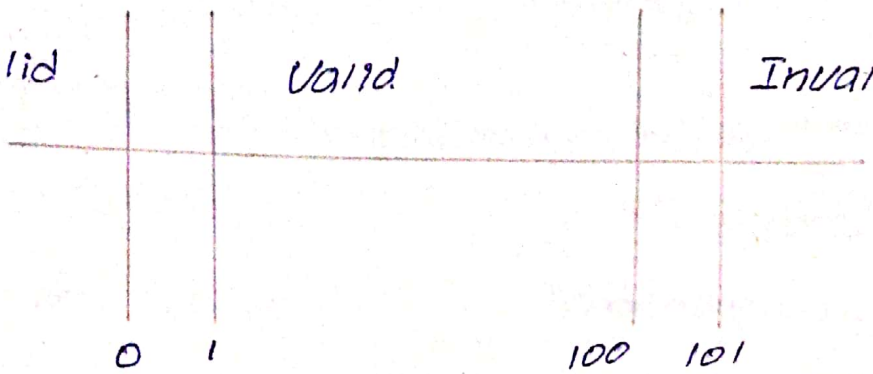
* Max, Max+1, Max-1
* 6, 7, 5

EXAMPLE :-

→ Program accepts 1 to 100 characters and identify boundary values

→ The valid inputs are: a-1, a, a+1, b-1, b, & b+1

→ 0, 1 to 100, 101



Scanned with CamScanner

DECISION TABLE :-

* A table showing combinations of inputs and outputs which can be used to design the test cases.

* This technique is also referred as cause effect table.

Decision table for Login Test :

	Rule 1	Rule 2	Rule 3
Conditions			
Valid user name	False	True	True
Valid password	-	False	True
Actions			
Login accepted	False	False	True

ATM Decision table :

	Rule 1	Rule 2	Rule 3	
Conditions				
Valid user name	False	True	T	T
Valid password	-	False	T	T
Adequate balance in account	-	-	F	T
Actions				
Login accepted	False	False	T	T
Amount transferred	-	-	F	T

Scanned with CamScanner

STATE TRANSITION TECHNIQUE :-

* Using state transition testing, we pick test cases from an application where we need to test different system transition.
* We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state.

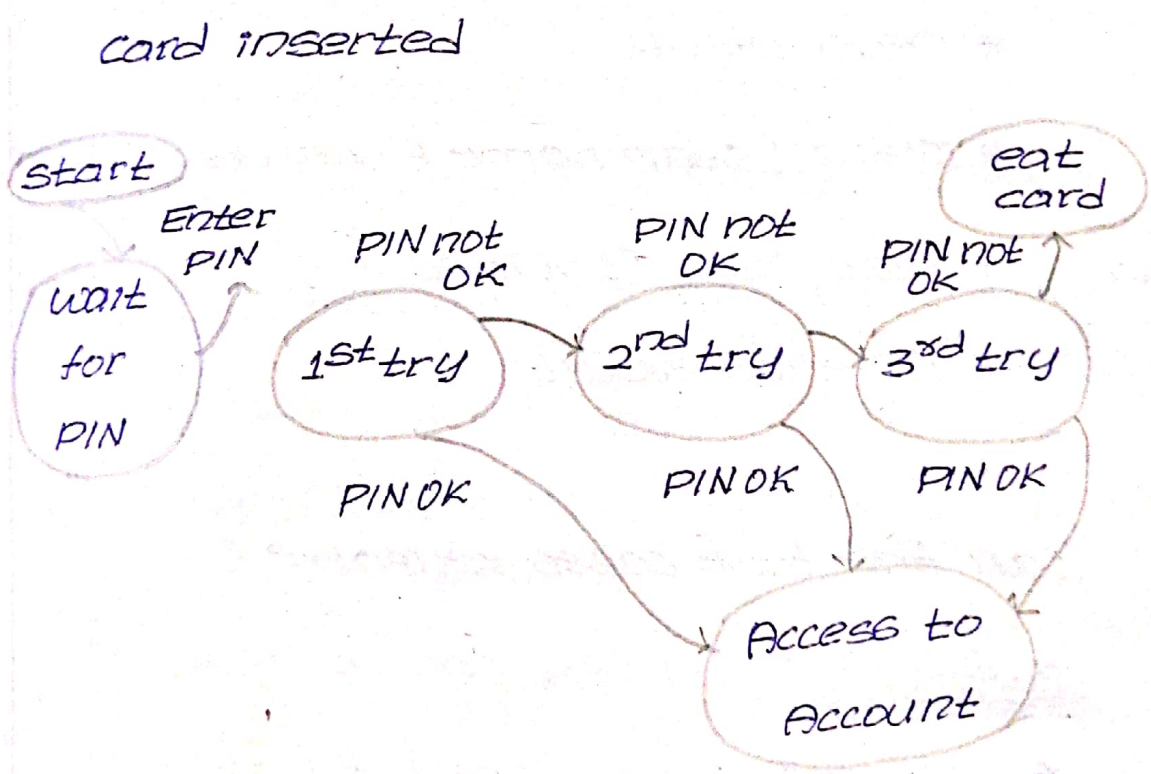


FIG. State transition technique

WHAT IS TEST CASE ?

Test case is a description of what to be tested, what data to be given and what actions to be done to check the actual result against the expected result.

WHAT ARE THE ITEMS OF TEST CASE ?

* Test case number
* pre-condition
* Description
* Title [step name & step description]
* Test Data [optional]
* Expected result
*

Can the test cases reusable ?

* Yes, test cases can be reusable.
* Test cases developed for functionality testing but it can be used for integration/ system / regression testing and perfo-rmance testing with few modifications.

Scanned with CamScanner

What are the characteristics of good test cases.?

* Test case should start with "what you are testing".
* Test case should be independent.
* Test case should not contain "If" statements.
* Test case should be uniform.

Are there any issues to be considered?

Yes there are few issues :-

* All the test cases should be traceable.
* There should not be too many duplicate test cases.
* Out dated test cases should be cleared off.
* All the test cases should be executable.

Scanned with CamScanner

USE CASE :-

Testing team follow usecase template inorder to design test cases.

Use case	Test case
* What are the steps tester need to follow inorder to test software feature.	* What are positive, negative action and what set of data use to operate software and what kind of output software should display we design as a test case.

EXAMPLE :-

Flipkart (Add to product usecase) :-

User Action	System Response
1. Enter Valid URL	* System should display Required page
2. Search product with Item name	* System should display Required product
3. Add product to cart	* System should Add product to cart.

Scanned with CamScanner

USE CASE TEMPLATE :-

Use case ID	Unique Use case name and ID for further usage
Use case Name	Use case name in referral document.
User story	Brief description about scenario what exactly user wants
Actors	primary: User, secondary: User
Description	Brief description of use case
Trigger	From where testable screen can launch
Preconditions	preconditions to follow before test application
Post conditions	post conditions to follow after test completed.
Normal flow	Original flow user access scenario in application

Scanned with CamScanner

Alternate flows	Alternate flow to access scenario from software
Priority	Medium
Business rules	1. Following business rules only helps to design test case. 2. Original requirement should be mention in the business rules.
Special requirements	New mode : Editable or Non-editable fields. Mockup list of scenario and entity diagram of fields. Field Name → Types of fields → Read and write permission on field.
Message [Error/ Status Msgs]	Message display when User Enter valid data Message display when user Enter invalid data.

Scanned with CamScanner

USE CASE TEMPLATE REGISTRITION :-

Use case ID :	UC#OL-UR-0001-UC-001 PRD#:
Use case Name :	User Registration
User story	Our software provide regis--tration and login feature to communicate through internett. Here end user fill registration form before Login.
Actors	Primary :→ Any User who has knowledge in internet & Software.
Description :	User need to fill personal details to complete registra--tion.
Trigger :	Click Sign up now link from homepage.
Preconditions :	• Make user at Homepage
Post conditions :	• Should receive registration message, After registration user can access outlook login and further modules.

Scanned with CamScanner

Normal flow	1. User Access from Home page
Alternative flows:	1. NA
Priority:	High
Business Rules:	1. Make sure user should not have account with same email Address. 2. First name and last name appear as your screen name in your account. 3. User able to create account with two domain outlook and hotmail. 4. Password should be encrypted 5. User can view password by click on view password icon 6. Retype password should match with previous password. 7. Should not enter any first name or last name during password entry. 8. Person should have more than 18 years during registration time

Scanned with CamScanner

9. Person can select gender before registration.

10. All fields are mandatory.

Special Requirements:

Sign up now → link

First name → Edit box → Min 4 Max 40, only Alphabets, Mandatory.

Last name → Edit box → Min 4 Max 40, Only Alphabets, Mandatory.

Email :→ Edit box :→ Min 3, Max 100, Allow Alphabets, Numeric, some special characters (. @ $ &).

Password :→ Edit box :→ at least 8 characters, Allow all characters.

Retype password :→ Edit box :→ match with password.

Gender :→ Radio button

DOB :→ List

Scanned with CamScanner

TEST CASE TEMPLATE :-

* Section
* Description
* Seenario ID
* Type of test
* Pre condition
* Test case ID
* Test case title
* Test Data
* Expected Result
* Actual result
* Status

TEST CASE TEMPLATE :-

* Test case Name / ID
* Featurename/ Reference ID
* Description
* Pre condition
* Input Field Name
* Step number
* Step description

Scanned with CamScanner

* Expected Result

* Actual Result

* Status

TEST DATA :-

* In general test data provided by the client. In some cases test Engineer generate test data [Dummy data] later testing team forward same data to client as deliverable document.

Test Data

Real Data

*Account number :-

Valid: 00123456789

Invalid: 90123421567

* SOAP → DOV
Liril

* Real data provided by client

Dummy Data

accepts only alphabets & numbers

* Fname

Valid: abcd, abcdefghijkl, abc1, abcd123abcd

Invalid: 1abcd, 12345, @12ab, abcdefghijklm, abcde123;234k

* Dummy data created by testing team.

Scanned with CamScanner

INSERTING DATA TO DATABASE :-

* Using API insert data into database

DATA GENERATOR :-

* TOAD : Toad is a automated Data generator. Using toad testing team fetch data from database test execution.

TEST CASE REVIEWS :-

After complete test designs different people involve in reviews to approve test design document.

1. Peer review
2. Test Lead review
3. Developer Lead review
4. Business analyst review
5. Test Manager review / Project manager review

Scanned with CamScanner

1. PEER REVIEW :-

Here one of your team member involve in review. During this review they concentrate on few things

1. Any spelling mistakes in testcases
2. Test designs are aligned
3. Test case order
4. Easy to understand

2. TEST LEAD REVIEW :-

Here test Lead involve in testcase review. During this test test lead consider few things.

1. Any missing testcases
2. completeness of test cases
3. completeness of test Data
4. correctness of test cases & testdata

3. DEVELOPER LEAD REVIEW :-

After completed test Lead review testing team forward test design document to development team Lead.

Scanned with CamScanner

Here development team lead concen-trate on

1. whether testing team followed all bylaws to design testcases.
2. Are any test case out of concept
3. Are any test case misleading

4. BUSINESS ANALYST REVIEW :-

After completed developer lead review Business Analyst concentrate on

1. Whether testcase able to test software as client expected
2. All prediction matching with client requirement.
3. Are test designs covered for all requirements.

5. PROJECT MANAGEMENT REVIEW :-

Collect all reviews and strategies from all team members and approve test design to continue execution.

Scanned with CamScanner

SOFTWARE DEPLOYEMENT INTO TEST ENVIRONMENT :-

After completed test design reviews and approvals. Testing team lead request development team lead to install soft-ware in testing environment (stagging environment).

offshore	onshore
Installation at local servers	Installation at client place

NOTE :-

In recent days organization maintaining seperate devops team to manage all deployments. This team monitor everyday development team work and create maintain & create different version of software builds.

* After Deployment software into stagging environment development release below document to testing team. So that testing team can verify Hardware / Software

Scanned with CamScanner

installations before proceeding System testing

1. Deployement Document

2. Release Note / Build Note

DEPLOYEMENT DOCUMENT :-

* Document ID
* Build Version
* Hardware :- 4GB RAM , Processors
* Softwares required :- OS [software
* Browser Version :- IE Browser
* Approvals

RELEASE NOTE / BUILD NOTE :-

* Document ID
* Build Version
* URL :- http://192.168.1.12/Home
* Authentication :- UID / PWD's
* DB-connecting :- IP Address

Scanned with CamScanner

TEST EXECUTION :

After successful installation of software into testing environment. Testing team start test execution on real build.

FORMAL TESTING :-

* Smoke testing
* Comprehensive testing
* Sanity testing
* Re-testing
* Regression testing
* System Integration Testing

INFORMAL TESTING :-

* Adhoc testing
* Exploratory testing
* Monkey testing
* Gorilla testing

Scanned with CamScanner

SMOKE TESTING :-

* Smoke testing conducting on initial builds.

* Whenever software receive from developer team, testing team involve in smoke testing to verify software stability.

* Inorder to verify software stability testing team check all the stability major and important functionalities working before continue to real execution (or) exhastive testing.

* Incase testing team feel software is not stable, testing team can suspend build and request for stable release.

Initial Builds

- Build #1
- Build #2
- Build #3
- Build #4

} Smoke testing

.
.
.

Scanned with CamScanner

modified Build

Build#10
Build#11
Build#12
} Sanity testing

SANITY TESTING :-

* Sanity testing conducts on modified software
* Sanity testing is similar to smoke test only difference is it is tested on modified software.
* After reporting defects, ~~to~~ development team fix defects and release new version of software after few modifications.
* Then testing team conduct sanity to check software stability after bug fixing.

TEST BED :-

Combination of environmental test case is called test bed.

Scanned with CamScanner

DIFFERENCE BETWEEN SMOKE & SINITY TESTING :-

Smoke testing	Sanity testing
* Also known as build verification testing	* Also known as
* Conducts on initial software when devel--opment team release software for the first time.	* conduct on initial software when devel--opment team releases software to testing after fixing few defects.
* Smoke is scripted (can design test cases).	* Sanity is unscripted (No Test Design)

COMPREHENSIVE TESTING :-

In development comprehensive testing means checking every line of code. In testing comprehensive means executing every single test case by operating software.

1. Select a test case
2. Read and understand detail (description & precondition).

Scanned with CamScanner

3. Read test case title and operate software according to steps.

4. Verify expected result match with actual result.

5. If expected result match with actual result. Make test case status as passed and then write actual result and continue test case by following order.

6. If expected result mismatch with actual result change status of testcase as failed and write actual result what exactly found.

7. After testcases failed, immediately report a defect to development team using available defect reporting tool.

8. Find failed dependent testcases and change testcase status from No run to Not. execution

9. Continue to all testcase execution using order.

Scanned with CamScanner

RETESTING :-

* In first build tester report a defect wherever found mismatch between expected and actual result.

* Those defects are fixed by the development team and release new version of software to testing team for evaluation of defects.

* During this time testing team concentrate on fixed defects.

* Then testing team retest only failed test cases against new build

Total number of testcase execution on firstbuild

100

Passed test cases	Uncovered due to dependent test case failure	Failed test cases Retesting Build to

Scanned with CamScanner

REGRESSION TESTING :

* Regression testing conduct on modified software same as retesting but only difference between regression testing and ret time testing team execute all failed test cases and some of bug passed test cases.

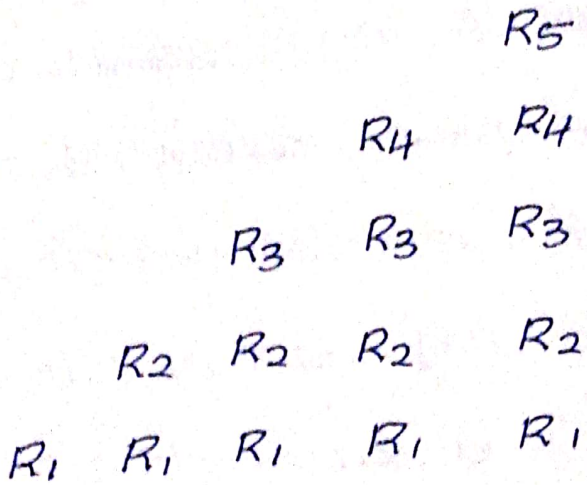
* Reason is while fixing time developer change program code, this modification may effects to existing passed test case as well so far that testing team re-execute passed testcases in previous build again agin on modified software.

When regression testing conducts :

* When software modified for big fixes.
* When software upgraded UI.
* When browser version changed.
* When new requirements introduced.
* When any feature removed from software
* Uncovered test case in previous build.

Scanned with CamScanner

Regression cycle :



* If something introduced new to software which means you are inventing new bug to software

Note :- Most of time testing team involve in regression testing once

TEST CASE EXECUTION :-

Execution and execution results play a vital role in the testing. Each and every activity should have proof.

The following activities should be taken care :

1. No of test cases executed
2. No of defects found
3. screen shots of successful and failure executions should be taken

Scanned with CamScanner

in word document.

4. Time taken to execute.

5. Time wasted due to the unavailabi-
-lity of the system

TEST CASE EXECUTION PROCESS:

Take the test case document

check the availability of application

operate software by executing
each test case

Raise Defect Any Mismatch Found

INPUTS:

* Test cases
* System Availability
* Data Availability

Process

* Test it

OUTPUT:

* Raise the Defect
* Take screen shot & save it

Scanned with CamScanner

DEFECT HANDLING :

WHAT IS DEFECT ?

In computer technology, a Defect is a coding error in a computer program. It is defined by saying that "A software error is present when the program does not do what its end user reasonably expects it to do.

Defect is a Mismatch between expected and actual behavior.

WHO CAN REPORT A DEFECT ?

Anyone who has involved in software development life cycle and who is using the software can report a defect. In most of the cases defects are reported by Testing Team.

A short list of people expected to report bugs.

- Testers / QA Engineer
- Developers
- Technical Support

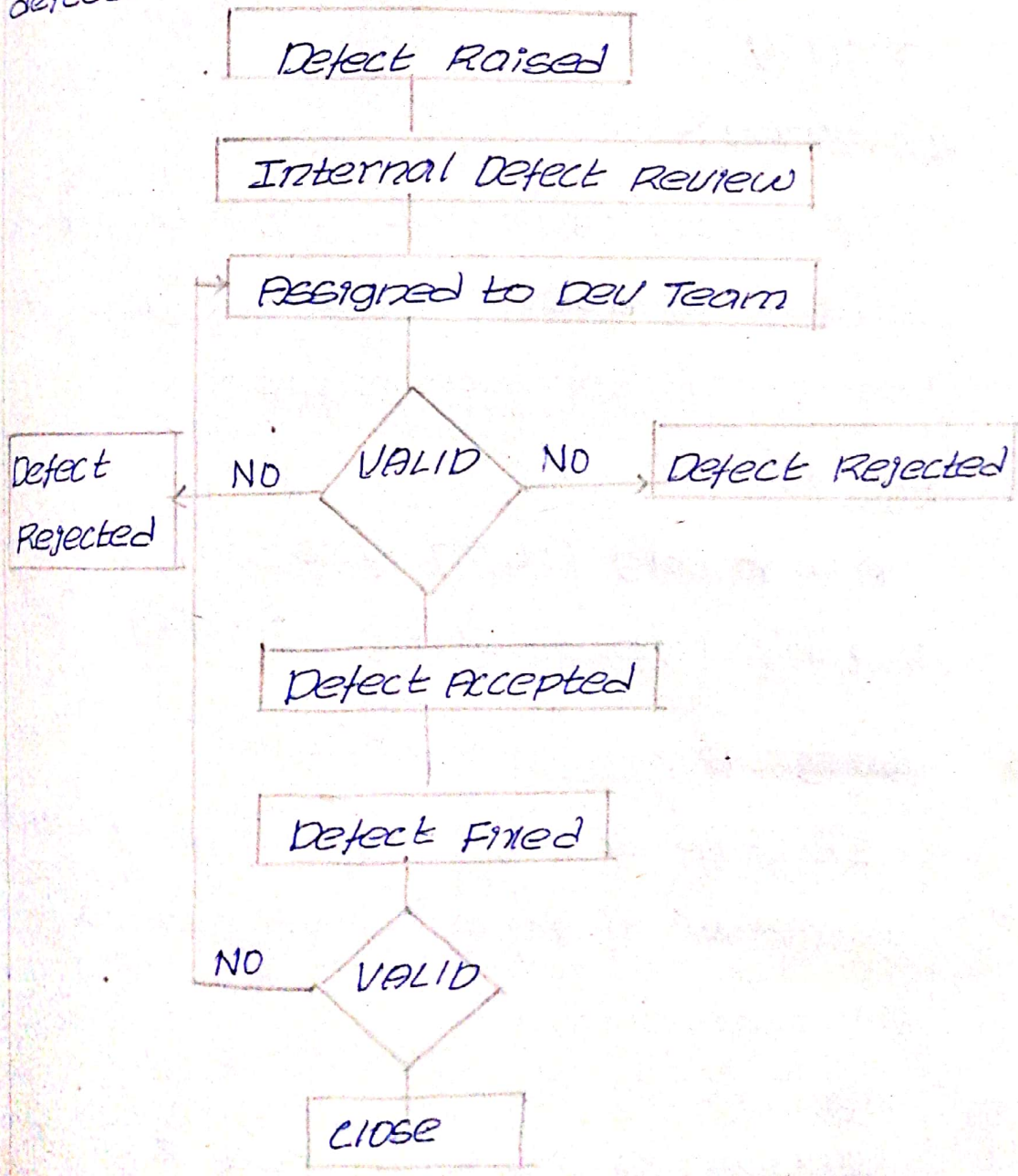
Scanned with CamScanner

End Users

Sales and Marketing Engineer

DEFECT LIFE CYCLE :-

Defect Lify cycle helps in handling defects efficiently. This DLC will help the users to know the status of the defect.



Scanned with CamScanner

DEFECT STATUS :

NEW :-

When defect was posted first time Testing team give status as NEW.

OPEN :-

After defect meeting team lead give status as open based on defect Priority.

DIFFERED :-

If defect was not prior to fix in current release. Then defect will be postponed to next releases.

FIXED :-

After bug fixing succesful developer change defect status to fixed.

REJECTED :-

In case of duplicate or silly defects developer change defect status to Rejected.

Reopen :-

In case of duplicate Test Engineer Reopen defect in case defect was not

Scanned with CamScanner

fixed.

CLOSED :-

After successful bug fixing testing team change status to closed.

TYPES OF DEFECTS :-

- Cosmetic flow
- Data corruption
- Data loss
- Documentation issue
- Incorrect operation
- Installation problem
- Missing Feature
- Slow performance
- System crash
- Unexpected Behaviour
- Unfriendly behaviour
- Technical defects

Scanned with CamScanner

DEFECT TRACKING SHEET :-

Defect No	Description	Origin	Severity	Priority	Status
Unique No	Description of bug	Birthplace of the Bug	Critical Major Medium	High Medium Low Minor cosmetic	Submitted Accepted Fixed Rejected postponed closed Reopen

Scanned with CamScanner

DEFECT SUBMIT :-

While submitting defect test Engineer to fill more details to understand defect stage.

Items to Submit :-

1. Defect ID
2. Subject
3. TestcaseName / ID
4. Steps to produce
5. Status
6. Severity
7. Priority
8. Reproducable
9. Type of defect
10. Releaseversion
11. Sprint version
12. Assignto
13. comments
14. Screenshot

Scanned with CamScanner

DEFECT ID :-

Defect Management tool generate Unique ID for each defects which helps for future reference.

SUBJECT :-

* Here tester need to specify what is the actual mismatch found at software
* Here tester should describe defect shot and understandable way.

TEST CASE NAME OR ID :-

* birth place of defect, Here test engineer provide test case name/ID which helps developer to understand origin of defect.

STEPS TO PRODUCE :-

Here tester need to fill detail navigat-ions to reach defect.

1. How I Tested
2. What I expected
3. What I found

Scanned with CamScanner

Status :-

To track defect tester and develop-ment provide different status like

New

Open

Rejected

Differed

Reopen

closed

Evaluation [Not confirmed]

* Testing team only responsible to provide status as New, Reopen, Fixed / closed.

SEVIORITY :-

Here test Engineer is responsible to provide seviourity of defect. Test engineer provide different seviority status

[It is subject to project/organization]

* Fatal
* critical
* Ungent
* High
* Medium
* Low

Scanned with CamScanner

FATAL :-

Tester provide fatal when he found show stopper defect.

Critical :-

Tester provide critical when he found any key requirement not working

[If there is no work arround).

Urgent :-

Tester provide Urgent when any environmental / Technical

High :-

Tester provide ~~Urgent when~~ High when important feature is missing and any data loss or data corruption related defects.

Medium :-

Tester provide when any functionality have incorrect

Low :-

Tester provide Low when he found cosmatic defects. [No risks].

Scanned with CamScanner

PRIORITY :-

Tester define priority while submit defect but it won't take as final. priority will be decided after defect review.

[Defect review is a meeting conducted by development and testing team]

Priority items :-

P1 → Where defects need to fix immed-
-iately [with in four hours]

P2 → Where defects need to fix within business day

P3 → Where defects need to fix before Sprint release.

P4 → Where defects need to fix before Shipment of product to stake holder

REPRODUCABLE :-

Here tester need to analyse this defect may occur in future release. tester only provide Yes/No status.

Scanned with CamScanner

TYPE OF DEFECTS :-

Tester submit defect type like Functional, Technical, validation, GUI, performance, security defect, Data loss, Data corruption, system crash, Database etc.

RELEASE VERSION :-

Tester should choose version name / number of the release [Tester can choose from listBox according to testplan releasenote].

SPRINT VERSION :-

Tester should choose version number/ name of sprint [Tester can choose from list box according to test plan/Release note].

Screenshot :-

Tester should take screen shot of every defect and attach to defect.

[In manual we have to store in word document].

Scanned with CamScanner

TEST METRIX :-

In daily Job testing team report to test Engineer

* EOD Report
* Daily Execution report
* Daily defect status Report
* weekly status report.

BASE METRIX :-

Based on testing team report, test lead prepare base metrix.

$$\frac{\text{Total no.of tc's executed}}{\text{no.of tc's to be executed}} \times 100$$

$$\frac{\text{Total no.of failed tc's}}{\text{Total no.of tc's executed}} \times 100$$

$$\frac{\text{Total no.of passed tc's}}{\text{Total no.of tc's executed}} \times 100$$

$$\frac{\text{Total no.of not executed tc's}}{\text{Total no.of tc's to be executed}} \times 100$$

Scanned with CamScanner

$$\frac{\text{Total no. of defects found}}{\text{Total no. of test cases executed}} \times 100$$

$$\frac{\text{Total no. of High seviority defect}}{\text{Total no. of Defects}} \times 100$$

$$\frac{\text{Total no. of Low seviority defect}}{\text{Total no. of Defects}} \times 100$$

WHY TEST METRIX :-

* In case of delay in a test execution organization can alert testing team.
* In case of delay in a delivery organization can alert client.

CALCULATED METRIX :-

Based on testlead report PM/TM create calculated Metrix.

Scanned with CamScanner