TESTING.

Module:: Manual testing.

1. Testing Concepts (Theory) What ?
2. Testing project (practical) How ?
3. Agile process - Jira.

What is Software

A software is a collection of Computer programs that helps us to perform a test.

To run the System → System → OS, Device Driver, Servers, Utilities

it does processes → programming → Compilers, debuggers, interpreters,

Users use the APPS → Application → Web Application, Mobile Apps, Desktop Application → MS word, calc, paint, Notepad

Application::

e Commerce,
Banking,
Insurance,
Travel,

Software testing:

X Bank Company ⟶ IT company.

Requirements, Budget, Time,

IT → Develope company requirement
→ Testing. (Deliver quality of product) intention.
→ Deployment. / Delivering.

1. It is a part of software development process

2. It is an activity to detect and identify the defects in the software.

3. Objective of testing is to release quality product to the client.

Software quality :

1. Bug - free. (100%. Won't, try 99%).
2. Delivered on time
3. Within budget.
4. Meets requirements & /or expectations.
5. Maintainable. (user friendly).

project vs product.

Serivece Based Companies → TCS, Accenture

product Based Companies. → Google, microsoft oracle.

Software Application is developed for:-

For specific customer based on the requirement is called project

For multiple customer based on market requirement is called product.

project:	product:
HDFC internet	Whatsapp.
~~Axis~~	MS office

Why do we need Testing

Ultimate goal is deliver quality product.

ensure, does it have bugs

customer requirement.

Error, Bug & Failure.

Error - Human mistake,
incorrect Human action.

Bug/ Defect - Deviation of expected & Actual behaivour.
Tester environment.

Failure - Installed Software on customer environment
After a few days, something not Working.

Why the Software has bugs.

Miscommunication or no Communication.
Software Complexity.
programming errors.
Changing requirements (Frequently).
Lack of skilled testers. (Testers Skill set).

∴ Testers :-
Break the application.
Negative approach.

SDLC / STLC :

SDLC → Software Developement Life cycle.

SDLC is a process used by Software industry to design, develope & test Softwares.

3'P → people, process, product.

1. Requirements Analysis
Collect Requirement from customer, understand project & product managers involve in the phase.

2. Design
Designers, design the software, UI, navigation, like

3. Development
Developing the software. by programming languages

4. Testing:
After completing development.
Conduct testing
functional
non functional.

5. Maintenance.
After Deployment, maintaining.

Some Important models:

Waterfall model:

1. Requirement Analysis:

:: collect or gather the requirement from customers. by project/ product managers.

They prepare the document is called SRS document. (Software Requirement Specification)

After Document prepared, it goes to next step.

2. System Design.

Design Document. phase.

3. implementation:

implementation or coding phase

4. Testing.

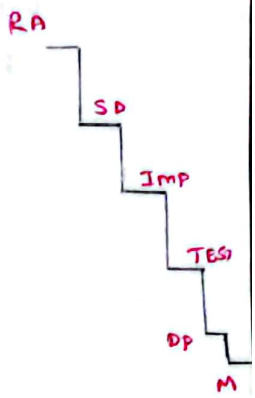
Testing | QA | QC.

5. Deployment.

Delivered

6. Maintenance.

Regular.



* Each & every step, which has input and it produce outcome.

Input ↓
Out Come. →

* Traditional model.

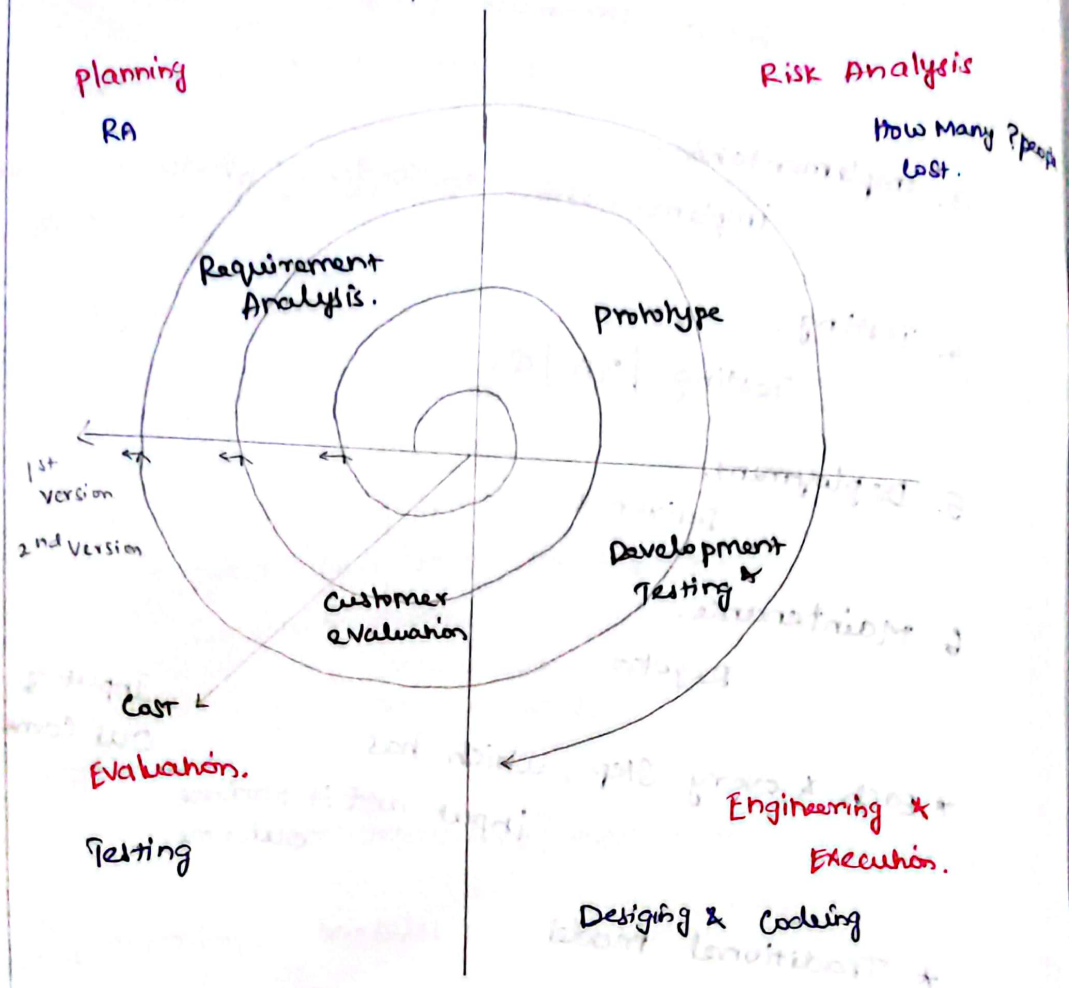
merits:

Quality good.
it cannot Change requirement, After processed.
less bugs, bcoz, no new requirements.
preferred small projects.

De-merits:

1. Requirement changes not allowed.
2. if there defect in requirement, that continue in all phases.
3. Investment more, time taking for rework
4. Testing will start only after coding.

Spiral Model : iterator model.



Its Accomodate new requirement.

every cycle, new requirement can added especially, product based companies.

Ex: Whatsapp
Windows
MS suits
Gmail.
} new version released, every cycles.

merits:

iterative model / Version control model
Requirements allowed,
new requirements depends on old model / cycle.
every cycle, new software released.
software released in multiple version.

demerits:

Every cycle testing done, b4 going to next cycle
every module customer will use software.

Demerits:

Requirement not allowed in b/w cycle.
every cycle looks like waterfall.
no testing in requirement / design phase.

prototype model:

Waterfall + Spiral :

Blue print of the software.

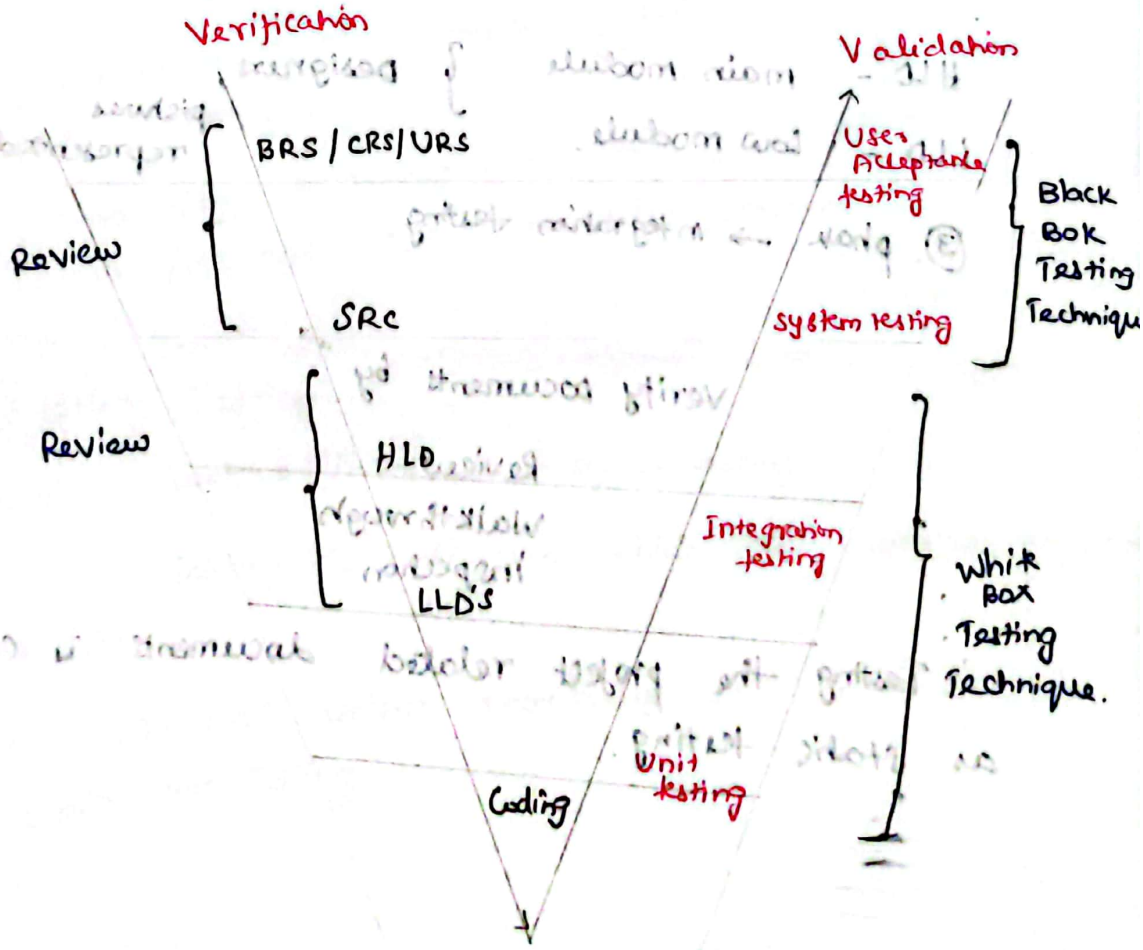
RA from customer → prototype (sample) → customer → design, coding, testing ....

∴ module is piece of software.

Software are contains multiple module :.

V model / VV model:

Verification / Validition :



BRS - Business Requirement Specification

CRS - Customer Requirement Specification

URS - User Requirement Specification

↑

Client / Business Unit

(BU) Business

Business people understand / not by technical people.

① phase → User Acceptance Testing

....

SRS →

Technical people understand

(PM) project manager

② phase → system testing

...

HLD - main module } designers,

LLD - Low module. pictures represented.

③. phase → integration testing.

Verify documents by

Review

Walkthrough

inspection

Testing the project related documents is called as static testing.

Unit testing:-

Testing the single module or component of the Software.

Developer conduct the unit testing.

After developing coding, unit testing done by devps. (bcoz he knows as them coding).

Integration testing:-

Integrated the module, main module.

After developed all module, integrated, then do integration testing.

Developers only test integration testing.

White Box testing

Whatever the testing conduct at code level, sub module or main module, it's called white box testing techniques.

→ Unit & integration testing, is WBT.

→ Internal logic of modules, / program

System Testing:-

Testers involved in system testing.

Testing the software, which meets customer requirements or not.

No need coding knowledge, so it is called Black Box testing.

User Acceptance Testing :-

Here, customer do testing along with testers.

This we say Black Box testing.

↑ Black Box testing :-

It is the technique of testing done by testers.

Functionality is working as per customer requirements

Verification → Static testing B4 software ready

Validation → Dynamic testing After software ready

Dynamic testing

Testing the actual software.

Unit testing
Integration testing
System testing
UAT.

Verification :-

Verification checks whether we are building the product right

Focus on documentations.

Verification typically involves. Reviews, walkthrough, Inspection.

Validation:

Software developed, whether we are building the product right

Take places after Verification completion.

Focus on software.

Dynamic testing → Unit, integration, System, UAT.

Advantage:

Testing involved in each & every phase.

Dis-Advantage:

Documentation is more
initial investment is more.

Review:

Conduct on documents to ensure correctness and completeness.

Requirement Review
Design Review
Code Review
Test plan review
Test cases review etc.,

Walkthrough:

It is a informal review
Author reads the document or code and discuss with peers.
It's not preplanned and can be done whenever required
Also Walkthrough does not have minutes of the meet.

Inspection:

Its a most formal review type
In which at least 3 - 8 people will sit in the meeting
1- reader, 2- writer, 3- moderator plus concerned.
Inspection will have a proper schedule which will be intimated via email to the concerned devops/tester.

QA, QC & QE

Quality Assurance
Quality Control
Quality Engineer

P - people - QC (tester)
P - process - QA (process related)
P - product

QA	QC
QA is process related	QC is the achial testing of the software.
QA focuses on building in quality	QC focuses on testing for quality
QA helps to preventing defects	QC is detecting defects
QA is process oriented	QC is product oriented
QA for entire life cycle	QC for testing past in SDLC

UNIT Testing:

A Unit is a single Component or module of a Software

It Conduct on a Single program or single module.

It is a white box testing technique.

It conducted by developer.

Techniques:

- Basic path testing
- Control Structure testing
  - Conditional coverage
  - loops coverage.
- Mutation Testing.

Integration Testing:

Integration testing performed between 2 or more modules

Integration testing focuses on checking data Communication b/w multiple modules.

Integrated testing is white box testing.

A → B → C modules

+ + Integration.

Types:

1. incremental Integration testing
2. Non-incremental Integration testing.

Incremental:

Incrementally adding the Modules and testing the data flow between the modules.

Two Approaches:

TOP DOWN ← Ensures the module added is the child of previous modules

BOTTOM UP. → Ensure the module added is the parent of the previous module

System Testing :-

Testing over all functionality of the application with respective Client requirements.

Its a Black Box testing technique.

After completion of component (unit) testing and integration level testing's we start system testing

Before Conducting system testing we should know the Customer requirements.

ST focused on below techniques. / aspects:

1. User Interface Testing (GUI / UI)
2. Usability Testing
3. Functional Testing
4. Non-functional Testing.

UAT - User Acceptance Testing :

After Completion of System testing UAT team Conducts acceptance testing in two level.

Alpha testing
↳ Testers environment.

Beta testing
↳ Live environment.

ST:

1. GUI Testing:

Graphical User Interface testing is a process of testing the User Interface of an application.

GUI includes all the elements such as menus, checkbox, buttons, colours, fonts, size, icons, content, and images.

GUI Checklist:

1. size, position, width, height of the element.
2. error msg getting displayed or not.
3. diff section of the screen.
4. font readable / not
5. In diff resolution, check the screen by zoom in/out.
6. Alignments of text, icons, buttons etc., proper place/not.
7. Colours of font
8. image Clarity
9. Alignments of images
10. Spelling correctness
11. Attractive / Irritative
12. Scroll bars according to the size of the page
13. Disabled field.
14. Colour hyperlink
15. Testing UI Elements like button, text, textbox, textarea, Checkbox, radiobutton, dropdown, links etc..,

Usability Testing:

Checks how easily the end users are able to understand and operate the application is called Usability testing.

It is user-friendly or / not.

Functional Testing:

Functionality is nothing but behavior of application.

1. Object properties Testing
2. Data base Testing
3. Error handling
4. Calculations/Manipulation Testing
5. Links Existence & Links Execution.
6. Cookies & sessions.

Object properties Testing:

Check the properties of object present on the Application. Ex: enable, disable, visible, focus, either one radio button, Check box single, /multiple, limits, Case sensitive.

Data Base Testing / Backend Testing

DML operation - Data Manipulation Language

Basically, it checks the communication b/w Client (UI) ⟶ Server,

Whether the Data stored correctly / retrieve correctly

Insert.
Update,
Delete,
Select,

Table & column level validation (Column type, Column length, number of Column...)

Relation between the tables (Normalization)

Functions

procedures

Triggers

Indexes,

Views

etc...,

Ex: downloading file, fetching something....,

Error Handling Testing.

Tester verify the error msg. while performing incorrect actions on the application.

User understandle / simple language.

Ex: incorrect Data.
Invalid User / password.

Calculation / Manipulations testing.

Testers should verify the Calculation.

Links:

Where exactly the links are placed - links existence

Links are navigating to proper page or not - links execution

Interal links

external links

Broken links.

Cookies and sessions:

Cookies are temporary files created by browser while browsing the pages through internet.

Sessions are time slots created by the server. Sessions will be expired after some time (if u are idle for some time).

Non - Functional Testing:

① performance — speed of the application.

(i). Load
(ii). Stress
(iii) Volume

Tools
J-meter
Load Runner.

Load: Gradually Increasing the load on the application then check the speed.

Stress: Suddenly increase / decrease the load on the application and check the speed.

Volume: Check how much data is able to handle by the application.

Virtual number of users added by special machine / tool.

It is done by specialists / Unique environment.

② Secure Testing.

How secure our Application.

1. Authentication – users are valid or not
2. Authorization/ Access Control – permissions of the valid user.

③ Recovery testing:

Check the system change to abnormal to normal.

Ex: Restore the pages.

④ Compatibility testing.

Version 1.0 installed the system over, the application installing Version 2.0.

∴ Ex: Update, Upgrades...

Forward Compatibility
Backward Compatibility.
Hardware Compatibility (Configuration testing).

⑤ Installation Testing:

Check screens are clear to understand.
Simple or not
Un-installation, all files get deleted or not.

⑥ Sanitation / garbage testing:

If any application provides extra features / functionality then we consider them as bug.

Validates functionality of software	verify the performance, security, reliability of the software.
Functionality describes what software does	Non-functionality describes how software works.
Concentrates on user requirements	Concentrates on user expectations
It takes place before non-functional testing	It perform after finishing functional testing.

Types of Testing:

Regression Testing:

Testing conduct on modified builds to make sure there will not be impact on existing functionality because of changes like adding / deleting / modifying the features.

Unit RT:

Testing only the changes / modification done by developer.

Regional RT:

Testing the modified modules along with impacted modules.

Impact Analysis meeting conducts to identify impact modules with QA & Dev.

FULL RT:

Testing main features & remaining part of the applica

Ex: if dev team done many changes on modules, when we perform one round of full regression.

Re-Testing:

once the developer fix the bug and we need to test again whether the bug is fixed or not.

If defect is not fixed, again raises the bug to devops.

Smoke Testing	Sanity testing.
Smoke test is done to make sure the build we received from the development team is testable / stable or not.	Sanity test is done during the release phase to check for the main functionalities of the application without going deeper.
Smoke testing performed by both developer and Testers	Sanity Testing is performed by testers alone.
Here build may be either stable / unstable	Here build is relatively stable.
It is done on intial build	It is done on stable build
It is part of basic testing	It is part of regression testing.
Usually it is done every time there is a new build release	It is planned when there is no enough time to do in depth testing.

Exploratory Testing:

We have to explore the application, understand completely and test it.

We do exploratory testing when the application ready but there is no requirements

Drawbacks:

misunderstand
Time waste,
Wrong defects identify.

AD-HOC Testing:

Testing application randomly without any test cases or any business requirement document.

Adhoc testing is an informal testing-type with an aim to break the system.

Tester should have knowledge of application even not having req / test cases.

usually Unplanned activity.

→ No documentation
→ No Test Design
→ No Test Case.

Monkey / Gorilla Testing:

Randomly testing Application
Do not have knowledge of application.
Gaming application suitable.

Positive Testing:

Testing the application with valid input is called as positive Testing.

providing positive inputs. Check expected result.

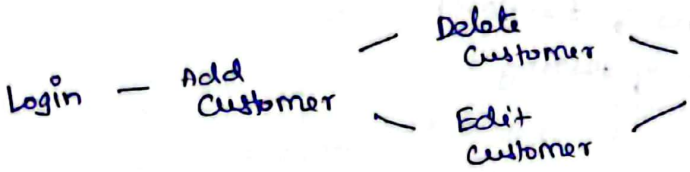
Negative Testing:

Testing the application with invalid input is called as Negative testing.

To Checks whether it behaves as expected with negative input.

End To End Testing

Testing the overall functionalities of the System including the data integration among all the modules is called end to end testing.



Globalization and localization Testing:

Application Support globally, by testing supports every language, currency format, mobile number, Date and time format. Ex: Facebook, paypal etc., (I18N Testing).

Application Support locally / Specific country, by testing local language, Date, currency etc.,
Ex: Chinese softwares

Test Design Technique:

→ Helps to design better cases
→ Used to prepare data for testing.
→ Different combination of data, prepare, it covers all functionalities.
→ Reduce / minimal the data / test cases.

1. Equivalente Class partioning (ECP)
2. Boundary Value Analysis (BVA)
3. Decision Table Based Testing
4. State Transition.
5. Error guessing.

Equivalence Class partitioning: (ECP)

Partition data into various classes and we can select data according to class and then test

Reduce no. of test cases
save time of testing.

Boundary Value Analysis (BVA)

BVA technique used to check Boundaries of the input

six test data : min, min-1, min+1
max, max-1, max+1

∴ ECP & BVA,
are Input domain testing

The Value will be verified in the text box / input field

Decision Table Example

Decision Table is also called as Cause-Effect Table.

This technique will be used if we have more Conditions and Corresponding actions.

To identify the test cases with decision table

Conditions:		Actions:	
Login	valid U/P	Homepage	✓
Login	V-U/IV-P	Homepage	No
Login	IV-U/V-P	Homepage	No
Login	IV-UP	Homepage	No

State Transition.

This technique Change the input condition and State of the Application.

providing Various inputs (pos/neg) to test values of system behaviour.

Eg:- Login Attempts, (like max three times, after locked)
ATM pin.

Error guessing:

It is used to find bugs in a software application based tester's prior experience.

Not any rules to follow.

Test Analytical skills and experience.

Eg:- Entering invalid values in alpha text box, enter numbers.
Submitting a form without entering values.

SOFTWARE TESTING LIFE CYCLE:

1. Requirement Analysis.
2. Test planning.
3. Test Designing
4. Test Execution / Defect Reporting & Tracking
5. Test Closure / sign off

STLC is defined as a sequence of activities conducted to perform software testing.

Each of these stages have a definite Entry & exit criteria.

1. Requirement Analysis

Entry : BR Document and Design Specification Document.

Understanding the requirement
Clarification of doubts in Walk through session
Identify types of test to be perform

Exit: project plan / Functional Requirements.

2. Test planning:

Entry: project plan / Functional requirement

Done by testing team lead
preparation of test plan doc for various types of testing
Test tool selection.
Test estimation
Team formation.

What
How
When
} To Test

Exit: Test plan document.

3. Test Designing

Entry : Test plan document.

Create test case document with test scenario, test case and test data.

Review test cases with peers or lead

Automation testers prepare skeleton scripts

RTM will be done by testing team lead
get Approval from BA

Exit: Test case Document, RTM..

4. Test Execution:

Entry: Test case Document, RTM

Execute test as per plan
Raise defects & Track the defects to closure
prepare defect report and test report / test log

Exit: Test Report and Defect Report.

5. Test Closure / Signoff:

Entry: Execution result / Defect log

Will be done by testing team manager
Test result analysis and bug report analysis
Evaluate cycle completion
prepare test Closure Report:

Exit: Test Summary Report / Closure Document.

Test plan template Contents:

Overview
Scope
inclusion
Test Environments / Test Bed
Exclusion.
Test strategy
Test schedule
Test Tools
Defect Reporting procedure.
Risks & mitigations
Approvals.

Use Case, Test Scenario & Test Case, Test Data.

Use Case:

Use Case describes the requirement provided by BA, Team manager.

Test Scenario:

It is high level Test Case. It is an idea of what we are going to test.

Test Case:

step by step achons to be performed to Validate functionality. It means how to test

Test Case contains test steps, expected results & actual results

Test Case Contents:

Test Case ID: (TC_01)
Test Case Title
Description
pre condition
priority
Test Data
Achons
Expected Result
Actual Result
Remarks / Result.

Requirement Traceability / Matrix (RTM).

RTM describes the mapping of Requirement's with the test cases.

Main purpose of RTM, to track all functionality which covered.

Parameters RTM:

- Req ID
- Req Description
- Test Case ID's

Test Environment: / Test Bed.

Test environment is a platform specially build for test case execution on the software product.

It is created by integrating the required software & hardware along with proper network Configuration.

Defects / Bug.

Any mismatched functionality found in a application is called as defect / Bug / issue.

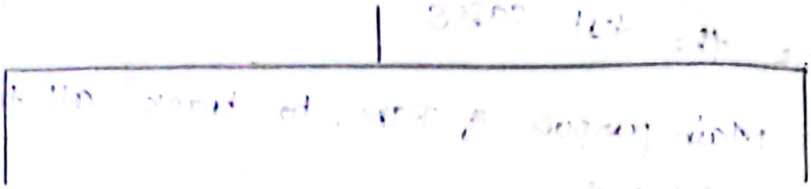
During test execution, test engineers are reporting mismatches as defect to developers through template or using tools:

Some Tools:

- Jira
- Clear quest
- Dev Track
- Bug zilla
- Quality Centre ... etc..

Defect classification:

Defects categorization.



Severity	priority.
Blocker	P1
Critical	P2
Major	P3
Minor	

Defect severity:

Severity describes the seriousness of defect and how much impact on Business workflow.

High - Application crashed,
Login not worked.

Medium - No confirm message, but process completed

Low : - Look & feel, overlaps, Spellings.

Defect priority:

priority describes the importance of defect.

How soon the defect is going to be fixed.

High (severity, priority) - login, search, payment etc.

High severity, low priority - links, websites etc..,

low severity, low priority - content, validation.

Low severity, high priority - logo, colour.

In real time, only categorized, priority as per we should fix the issues.

Defect Resolution:

Bug raised by testers, Devops team taking resolution like below

- Accept
- Reject
- Duplicate
- Enhancement
- need more information
- fixed
- As Designed.
- not Reproducible.

Dev.
accept the Ready to fix
reject.
already raised
not exactly defect, new feature or coming Version
screenshot,
- Actual designed, function their environment, not show

Defect / Bug life cycle:

Defines, State of a defect,

New:
When the defect is found and posted for the first time.

Assigned:
Once the defect is posted (bug), it can be assigned to developer or testing team lead, who will assign it to developer.

Open:
Once the developer opens the bug and check whether it is valid or invalid.

Fixed:
When developer fix the bug with appropriate codes.

Pending Retest:
Once the developer fixed the bug, it will be moved to testing team.

Retest:
When the tester retesting wether the bug raised was fixed or not.

Verified:
If the tester confirms that the bug is fixed.

Re open:

Bug not fixed,
Again, it reopen assigned devops.

Closed:

Bug is Fixed
Tester Close the raised defect status.

Duplicate:

If the bug is already raised by someone, again it raised means it is duplicate.

Reject / not a bug:

Devops think not a bug, or it is a enhancement

Deferred:

The bug which developer feels that it can be fixed later on the upcoming sprints / releases.

∴ In case, upcoming build, same defects are came. When we need to raise the new defect.

Test Cycle Closure:

Stop our testing.
When All TC's are passed, All defects are fixed
May have very minor bugs, one or two, BA should approve
BA, that conduct meeting, review all about testing periods what we have done.
outcome, deliverables, Test summary reports.

Test Metrics:

Tasks are to be tracked.

1. Test cases metrics.
2. Defect metrics.

∴ Defect Density. = $\frac{\text{no. of defects found}}{\text{no of requirement}}$

∴ Defect Removal Efficiency (DRE).

(A / A+B) * 100

$\frac{\text{Fixed defects}}{\text{fixed defects + missed defects}} * 100$

∴ Defect leakage:

$\frac{\text{no. of defects found in UAT}}{\text{no. of defects found in Testing}} * 100.$

∴ Defect Rejection Ratio:

$\frac{\text{no. of defect rejected}}{\text{no. of defect raised}} * 100$

∴ Defect Age: Fixed Date − Reported Date.

∴ Customer Satisfication: No. of Complaints per period of time.

AGILE METHODOLOGY.

Agile model | process:

It is an iterative and Incremental Approach.

Agile Methodology is a practice that promotes continuous iteration of development and testing throughout the software development life cycle of the project.

Here both development and testing activities are concurrent Unlike the waterfall model.

Why we go for Agile:

More Control
Better quality and productivity.
Higher return on investment.
Higher customer satisfaction.
Customer no need to wait till the complete software.

Agile Manifesto (principle):

1. Customer no need to wait for longtime.
2. We develop, test and release piece of software to the customer with few number of features.
3. We can accept | accomodate requirement changes.
4. Customer collaboration is good.

Advantages:

There will be good communication between Customer Business, Analyst, Developers & Testers.

Requirement: Changes are allowed in any stages of development.

Release will be very fast (weekly).

De-merits:

less focus on design & documentation since we deliver software very faster.

SCRUM:

Agile - is a process model.

Scrum - is a framework. helps to Agile to follow the principles.

Scrum is a framework through which we build software product by following Agile principle.

Scrum Team / Agile Team:

1. product owner.
2. Scrum Master
3. Dev Team
4. QA Team.

Scrum includes group of people called as Scrum team. Normally contains 5-9 members.

product owner:

Define the feature of the product.

prioritize features according to market value.

Adjust features & priority every iteration, as needed.

Accept or reject work result.

Scrum Master:

The main role is facilitating and driving the agile process.

It is a specific role. Skill set on Agile model.

Developers and QA:
Develop and Test the software.

Scrum Terminology

User story : A feature / module in a software.

EPIC : Collection of user stories.

product backlog : Contains list of user stories, prepared by product owner.
It's form of file xl, Word.

Sprint / iteration : period of time to complete the user stories, decided by the product owner and team, usually 2-4 weeks of time.

Sprint planning Meeting :
Meeting conduct with the team to define what can be delivered in the sprint and duration.
B4, sprint starts it conducts.

Sprint backlog :
List of committed stories by Dev / QA for specific sprint.

Scrum Meeting / scrum call / standup call :
Meeting conducted by scrum master everyday
15 mins. Discuss about : what status.
task complete till yesterday
What today plan and next plan
What are blockers

Q/A :
yesterday
Today
impediments / blockers.

Sprint Retrospective Meeting:

Conducts meeting after completion of Sprint
The entire team, including product owner
and scrum master participates.

After completion, sprint.

→ What went wrong, Well,
→ What new implements we have to do

One time conducted meeting.

STORY POINTS:

Rough estimation of User Stories, will be given by Dev & QA in the form of fibonacci series.

story points → 1, 2, 3, 5, 8, 12 ....

1 story point = 1 hour / 1 day (6 hours).

Login → Dev - 5, QA - 3

Burn Down Chart:

Shows how much work remaining in the Sprint. Maintained by the Scrum master daily.

Sprint review:

After completion of every sprint, Team will demonstrate the work to the product owner.

Duration - 1 hour.

AGILE:

Roles:
- product owner.
- Scrum Master
- Team.

Artifacts:
- product Backlog
- Sprint Backlog.
- Burndown chart.

Ceremonies:
- Sprint planning
- Daily scrum
- Sprint Review.

Scrum Board:

Stories	ToDo	inprogress	Testing	Done.

Agile Management tool - JIRA,

B4. Templates used like XI, word, pdf.