Software Quality Assurance(Manual Testing-Theory)

What is software?

Ans: Software is a collection of computer programs that helps to perform a task.

Type of software:

- System software: System software is used to run the system. Like device driver, operating system, etc.
- Programming software: Programming software is used to process the input data and make it an output. Like, compiler, debugger, interpreter, etc.
- Application software: Web application, Desktop application, Mobile application.

What is software testing?

Ans: Software testing is an activity that detects and identifies defects in the system.

What is quality software?

Ans: Software quality is the justification of software from the customer.

Parameter of quality software(When a software will be qualitiful)

I. Bug-free
II. Delivered on time
III. Within Budget.
IV. Meet Requirement
V. Maintainable

Product vs Project

Project: When a software application is developed for a specific customer based on the requirement than that is called Project.

Example :

Product: When a software application is developed for multiple customer based on the market requirements then that is called Product.

Example: Microsoft Word.

Why do we need testing?

Ans:

✓ We need testing to produce quality software.

- ✓ Then ensure that the software is bug-free.
- ✓ Ensure that the system meets customer requirements.

Error: Any incorrect human action that causes a problem in the system is called an error.

Bug/Defects: A mismatch of expected behavior and actual behavior of the product is called Bug/Defect.

Example:

Failure: Lack of customer expectation after release the system is called failure.

1) **Quality Assurance Vs Testing:**

(QA)	(QC)
❖ QA ensures the whole implementation process of the software and verification of the developed software.	❖ Qc ensures whether there is any error/bug/defect in the software or not.
❖ QA focuses on the process and procedure rather than actual testing	❖ Qc Focuses on the actual testing
❖ QA is Process Oriented Approach	❖ QC is Product Oriented Approach.
❖ QA Prevents the defects	❖ QC detects the defects.

SDLC Model (software development life cycle)

What is the SDLC model?

Ans: SDLC is a process used by the software industry to design, develop, and test the software

Purpose of SDLC : The purpose of SDLC is to produce software with the highest quality and lowest cost, within a shortest period of time.

SDLC model

- ✓ Waterfall model
- ✓ Spiral model
- ✓ V-model
- ✓ RAD model
- ✓ Big Bang model

SDLC Approach--(PADIT-M)----1) Planning 2) Analysis 3) Design 4) Implementation 5) Testing and Integration 6) Maintenance.

1. **Planning:** This is the first phase of the SDLC Model. In this phase, an organization needs to make a plan to produce a high-quality product.

2. **Analysis:** The project manager and Business analyst set up a meeting with the customer to collect all the information for the project and analyze what actually the customer wants to do, what is the purpose of the project and who will be the end-user.

Requirement Specification: In this phase, all the information gathered into the SRS documents as input then the software architecture uses that SRS document to implement that into the project

3. **Design:** In this phase, software architecture follows the SRS documents and transforms all the gathered requirements into a blueprint to develop a system software.

4. **Development/Implementation:** After the completion of the design phase, the developer starts coding and translating the software design into source code.

5. **Testing**: After the completion of the implementation phase, testing starts and modules are released for testing.

6. **Deployment** After the completion of testing, software is deployed in the customer environment.
   Maintenance After the deployment of that product on the Production/Customer/User environment, developers are responsible for fixed of any issues comes up.

Waterfall Model:

The waterfall model is the earliest SDLC approach. It is the oldest and most straightforward model. It is also known as the linear sequential model because the next phase starts only after the completion of the previous phase.

Advantage	Disadvantage
1. Simple and easy to understand, also	1. Difficult to understand and measure

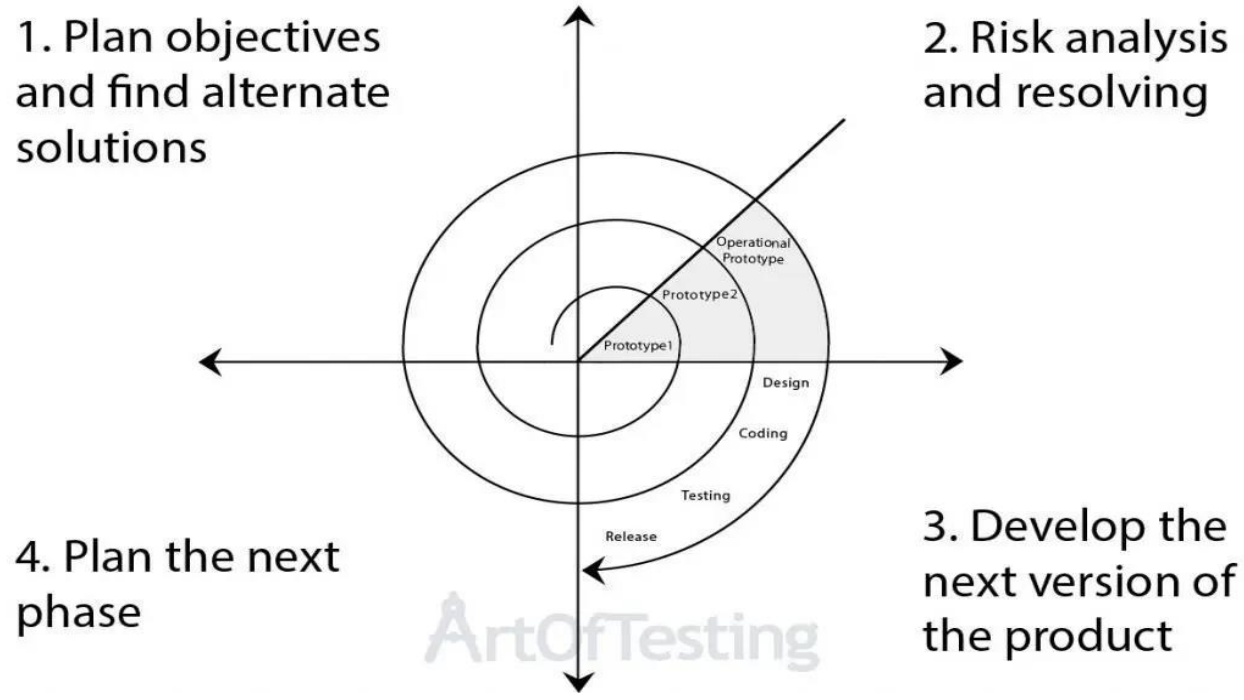
use	the progress
2. Bug will be less because requirement changes are not allowed.	2 Requirement changes are not allowed.
3. An appropriate model for a small project	3 An appropriate model for a complex project
4. Easy to arrange the tasks	4 (No backtracking) Difficult to change the customer's requirements
5 Process and result are well documented	

Process:-Anlysis---Requirement Specification----Design---Development----Testing – Deployment

Spiral Model

Spiral model is one of the most important SDLC models. It's a combination of an iterative development process model and a linear sequential development model.

The spiral model is preferred for product-based companies. Product > Product is software which is developed for multiple customers based on the market requirements.



Note: In the spiral model new version of software is released in every single cycle . Thus spiral model released multiple version of software in every cycle that's why it's called the version control model.

Advantages:

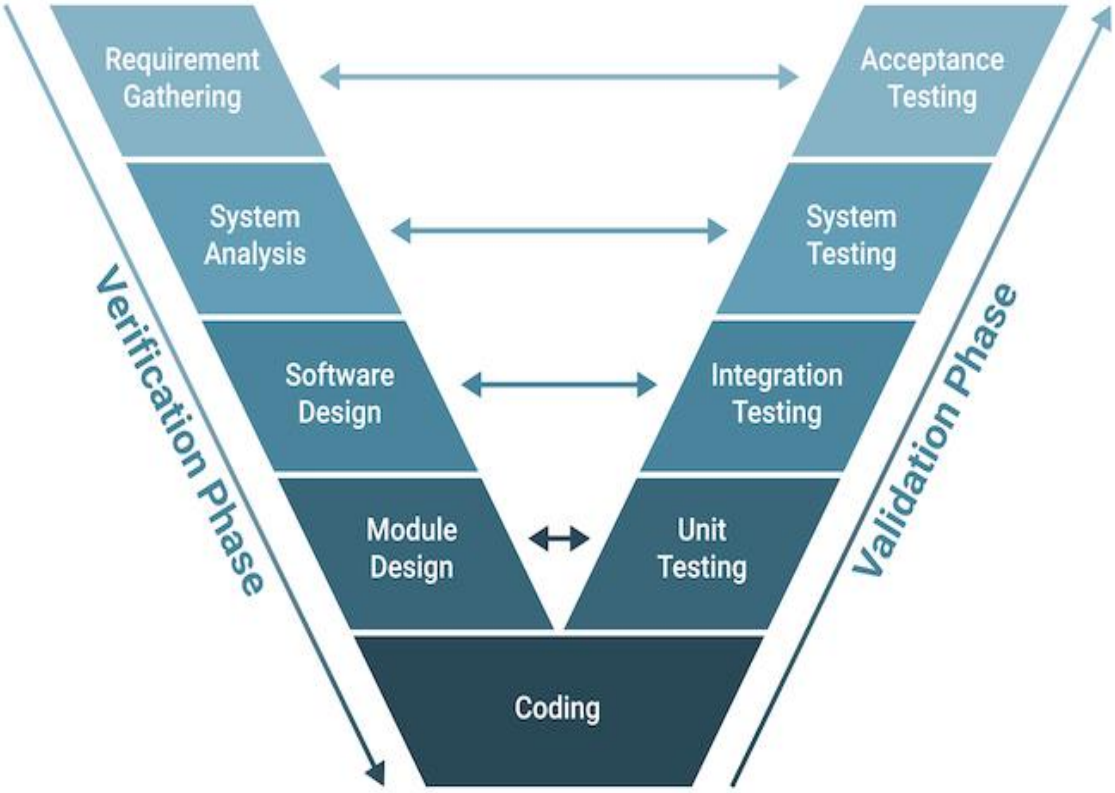
I) Customer don't need to wait for the completed software. because every single cycle a new version of the software is released
II) Testing is done in every single cycle.
III) Requirement changes are allowed.

V-Model

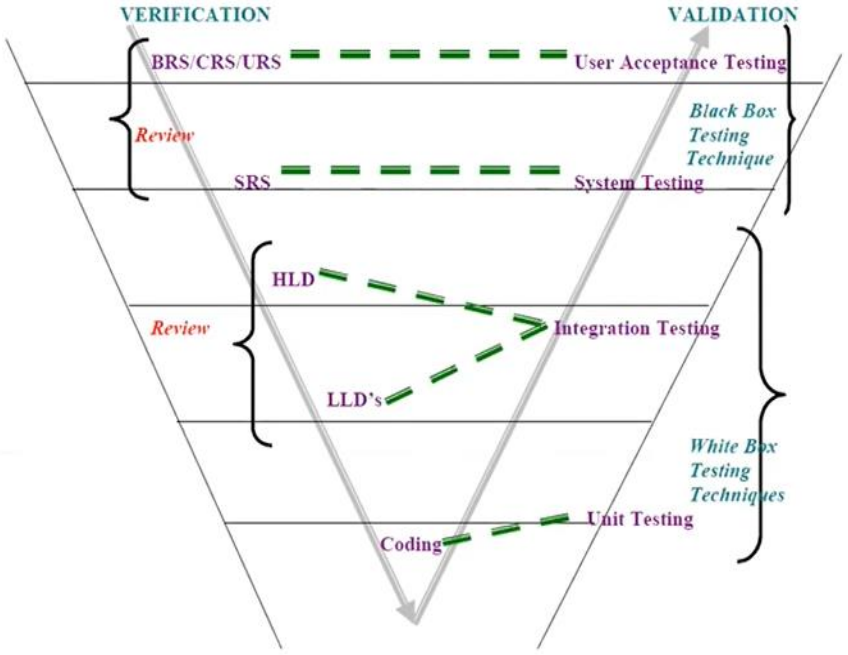
V-Model is an SDLC model where the execution of the process happens sequentially in V-shape. It is also known as the verification and validation model. V-model is an extension of the waterfall model.

Verification: Verification is the process of checking that software achieves it's goal/ specifications. It is also known as static testing. (Just testing without executing the code)… Verification happens only during the development phase.

Validation: Validation is the process of checking that software achieves the User's requirements. It is also known as dynamic testing (Dynamic means code executing)… Validation happens only during the testing phase.


Requirement Gathering
System Analysis
Software Design
Module Design
Coding
Unit Testing
Integration Testing
System Testing
Acceptance Testing
Verification Phase
Validation Phase

V-Model



1. BRS/CRS/URS-------Business analyst is responsible
2. SRS----------Project manager is responsible
3. HLD---------High level document.

   ----------Designer is responsible

   LLD----------Low level document.
4. Coding--------Developer.

SRS---Software Requirement Specification

Brs – Business Requirement Specification

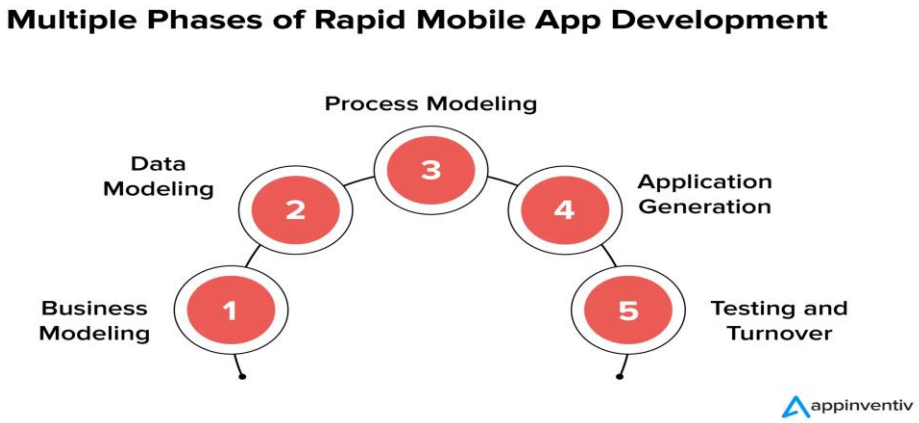
CRS ---Customer Requirement Specification

URS---User Requirement Specification

<u>Prototyping:</u> Prototype means sample. A prototype is a blueprint of a software.

RAD-Model

Rapid Application Development is a software development process that prioritizes on prototyping and quick feedback with user involvement. The rad model focuses on Time-Boxing rather than planning. A project should be completed in a short time.



When can you use the Rapid Application Development methodology?

I. **When user can reliably test your prototypes.**
When users can give consistent and reliable feedback on the prototypes to, then rapid application development is a great model to follow.

II. **When you've got the budget**
Compared to other development models, rapid application development is relatively inexpensive**,** but budget needs to be divided among the staff to build a project.

III. **When you need a project done quickly.**

If you've got a tight deadline, rapid application development is the best.

(Source:)

Agile model

An agile model/methodology is an iterative and incremental model.

=>Iterative means the same kind of process will repeat again and again.

=>Incremental means some features will be added to the existing software during the implementation process according to the customer's requirements.

Agile principles

- ✓ Customers don’t need to wait for a long time.
- ✓ Break down the project into small parts and release it to the customer.
- ✓ Requirement changes are allowed in any stage of software development.
- ✓ Delivery will be very fast within two weeks.

Agile manifesto

1) Customer satisfaction.
2) Welcome changes.
3) Faster delivery.
4) Good collaboration.
5) Simplicity.
6) Monitoring/Reviewing regularly.

Advantages	Disadvantages
✓ Customers don’t need to wait for a long time.	Less documentation in agile model due to early delivery.
✓ Break down the project into small parts and release it to the customer.	
✓ Requirement changes are allowed in any stage of software development.	
✓ The release will be very fast(within 2/3 weeks)	
✓ Good communication between teams.	
✓ It is very easy to adopt.	

Scrum : Scrum is a framework through which we develop a software by following Agile principles.

Scrum includes a group of people that is scrum team .

Scum Team:

- ✓ Product owner
- ✓ Scrum Master
- ✓ Dev team and QA team

Role of these Team:

Product Owner**:**

- Define the features of the product
- Prioritize the features according to the market value.
- Accept or reject the works result

Scrum Master: The main role of scrum master is facilitating and driving the agile process.

Dev and QA: Dev performs to develop the software and QA performs testing **.**

Scrum Terminology

User story: User story is a model/ feature in the software.

Epic: Epic is a collection of user stories. Epic represents functionality .

Product backlog: Product backlog contains the list of user stories which is prepared by product owner.

Sprint: Sprint is a period of time to complete the user stories which is decided by the product owner and the team.

Sprint planning meeting : Sprint planning meeting conduct with the team to define what can be delivered in the sprint.

Sprint backlog : Sprint backlog is a list of committed stories by the Dev/QA for sprint.

Scrum Meeting: Scrum Meeting conducted by the Scrum Master in everyday at least 15 mins is called Scrum call/ standup meeting.In scrum meeting these question will be must asked

What did you yesterday?

What will you do today?

Are there any backlog in your task?

Story Point: Story point is a rough estimation of user stories which will be given by Dev and QA in the form of Fibonacci Series.

1 Story Point = 1 hour/1 day (6 hours)

Login -→ Dev—5 h QA –3 h = 8 hours /1 day.

Burndown Chart: Burndown Charts shows how much work remaining in the sprint . Maintain by the Scrum Master daily.

Role:

Product owner: Product owner will define the requirements and write the story.

Scrum Master: Organize the meeting and drive the process.

Team : Will perform the task like development and Qa activities.

Arteffect: Arteffect means what are the documents we have.

- ✓ Product backlog
- ✓ Sprint backlog
- ✓ Burndown chart

Ceremonies :

- Srint Planning
- Daily Scrum
- Srint Review.

Static Testing vs Dynamic Testing

Static testing: Static testing is a type of software testing in which a software application is tested without executing the code. Static testing is performed in the early stage of software development to avoid errors. It is also called verification testing. ------

(Project related documents are tested)

Example: BRS/CRS/URS------

Static testing techniques:

- ⇒ Review
- ⇒ walk-through,
- ⇒ inspection,

Review: A review is conducted on the documents to ensure their correctness and completeness.

Note: Review can be done by anyone like dev or tester

Different type of review:

- ✓ Requirement review
- ✓ Design review
- ✓ Code review
- ✓ Test plan review
- ✓ Test case review

Walk-Through: Walk-through means informal review. After completing the review of yourself, 2/more people does the review that is walk-through.

Inspection: Most formal type of review. Three types of people involved in this review, those are reader, writer, and moderator.

Dynamic Testing: Dynamic testing is a type of software testing in which software application is tested with executing code to analyze the dynamic behavior of the code. It is performed at the later stage of the software development. It is known as validation testing.----- (Actual software is tested)

Dynamic Testing Techniques: For dynamic testing,

- ⇒ unit testing,
- ⇒ integration testing,
- ⇒ system testing,
- ⇒ user acceptance testing

Static Testing	Dynamic Testing
It is performed at the early stage of the software development.	It is performed at the later stage of the software development.
Static testing prevents defects	Dynamic testing finds and fixes the defects
Static testing is less costly	Dynamic testing is costly.
Static testing does the verification process	Dynamic testing does the validation process
Static testing can be performed before the compilation	Dynamic testing can be performed after the compilation.

Level of testing:

- ❖ Unit testing
- ❖ Integration testing
- ❖ System testing
- ❖ User Acceptance testing

Unit Testing(Level-1) white-box testing technique.

Unit Testing: Unit testing is a type of software testing where a particular module/component is tested by the developer is called unit testing.

Example: E-mail-------Login-----single component

Compose------Single component

Sent----------Single component

What is unit?

Ans: A unit is a single module or component.

Why unit testing is a white box testing?

Ans: Because the tester has to know the internal logic/code of the program.

Unit testing techniques:

I. Basis path testing
II. Control structure testing
III. Mutation testing
IV. Condition coverage
V. Loop Coverage

Basis path testing: Each and every line of the code should be executed at least once.

Control structure testing: control structure testing is used to increase the coverage area by testing various control structures.

Mutation testing: Mutation testing is a fault-based testing technique where we insert errors purposely into a program (under test) to verify whether the existing test case can detect the error or not.

Condition coverage: Varify the condition whether it is true or false.

Example: a=20, b=10

If a>b

Print(“a” islargest)

Else

“B” is largest

Loop coverage: Loop is a block of statements that can be repeated multiple times based on a certain condition.

In loop coverage , we will verify whether it starts from the beginning or not till the max.

Example: Print numbers from 1 to 10

```
#include <stdio.h>
int main() {
int i;
for (i = 1; i < 11; ++i)
{
printf("%d ", i);
}
return 0;
}
```

Integration Testing(Level-2) white-box testing technique

Integration Testing is a combination of individual units that is tested as a group is called Integration testing.

Example:

Integration testing features:

I. Integration testing performed between two or more modules.
II. Integration testing focuses on checking data communication between multiple modules
III. Integration testing is a white box testing technique.

Type of integration testing:

- Incremental testing
- Non-Incremental testing

Incremental module: Incrementally adding the module and testing the data flow of the module

Incremental Approach:

- Top-Down
- Bottom-up

System testing(Level-3)

System testing is a type of software testing that evaluates the complete or integrated software.

There are different types of system testing:

I. GUI- Graphical user interface testing
II. Usability testing
III. Functional testing
IV. Non-functional testing

I. **GUI- Graphical user interface testing**: Gui testing is a process of testing the user interface of an application.

Gui testing has two parts such as i) Frontend ii) Backend

Elements of GUI testing: Menus, checkboxes, buttons, colors, font, size, etc.

Gui testing checklist:

1. Testing the size, position, width, and height of the elements.
2. Check whether the error messages are displayed or not.
3. The font of the text is readable or not.
4. Alignment of the texts are in the proper place or not.
5. Testing the spelling mistake.
6. Checking whether the interface is attractive or not
7. Testing the UI elements.
8. Testing the colors of the hyperlinks.

II. **Usability testing:** Usability means easiness of using the software application. Checking how easily the end users can understand and operate the application is called usability testing.

III. **Functional Testing :** Functional testing is a type of software testing that is used to verify the functionality of the software application, and whether the function is working properly according to the requirement specification or not.

Functionality is nothing but the behavior of the application. Functional testing talks about how your features should work according to the customer's requirements

Example: To prevent the speed of a car, we need to know how to use a break-pad that is Functional Testing.

On the other hand, how the break-pad works is non-functional testing.

As a part of functional testing, these kinds of testing have to be conducted:

- Object properties testing
- Database testing
- Error handling
- Calculation and Manipulation testing
- Link existence and executions

- Object properties testing: Testing the properties of an object are working properly or not . Object means elements like buttons, text boxes, icons, images, etc.

Example :

(Object) Icon ----------→ Enable or Disable (Property)

(Object) Image---------→ Visible or Invisible (Property)

- Database testing: Database testing is a type of software testing that checks the data integrity, table, triggers, and scheme.

Why is Database Testing Important?

- ❖ Ensures database efficiency: Database testing helps to ensure the database's efficiency, maximum stability, performance, and security.

- ❖ Ensures information validity: Database testing helps to ensure the data values, formation received, and stored data in the database are valid or not.

- ❖ Helps to save data loss: Database testing helps to save data loss

- Error Handling: The Tester verifies the error messages while performing incorrect action.
  Error messages should be specific and readable

- Calculation and Manipulation Testing: The tester should verify the calculation with respect to requirements.

Whenever we see some calculations, we'll check that they are properly working or not with different sets of data.

- Link existence and execution: We can see links in the web application.

Link existence: Where exactly the links are placed according to requirements that is called link existence.

Link execution: Links are navigating to the proper page or not that is called links execution.

3-types of link

i. Internal link : when we click on a link that will navigate to the same page in a different section that is an internal link.
ii. External link: when we click on a link that will navigate to a different page That is an external link.
iii. Broken link: When we click on the link but it doesn't navigating anywhere.

Cookies & Session:

Cookies: Cookies are temporary files that created by the browser while browsing a page through internet is called cookies.

Session: A session is a time slot which is created for transactions/activities. Session is performed to ensure security .

Ex : After successfully login , if you do not do any action within some seconds the session will be expired.

Non-functional testing: Non-functional testing is a type of software testing that is used to verify the non-functional requirements of the application like performance, load ability, security, etc is called non-functional testing.

Non-functional testing focuses on customer's expectations.

Example:

Banking application	
Login (Functional)	Good / Bad Performance (Non-functional)
Fund transfer (Functional)	App is secured/not secured (Non-functional)
Check Balance (Functional)	Response fast/slow (Non-functional)

Types of non-functional testing: (PSRCCIS)

1. Performance testing: i) Load testing ii) Stress testing iii) Volume testing
2. Security testing
3. Recovery testing
4. Compatibility testing
5. Configuration testing
6. Installation testing
7. Sanity testing

1. Performance testing: Performance testing is a type of software testing that ensure the performance of the application under the workload.

Load testing: Gradually increase the load on the application and check how the speed of the application is.

Stress testing: Suddenly increase the load on the application and check how the application is responding.

Volume testing: Check how much data is able to be handled by the application.

2. Security testing: Security testing is a type of software testing that is used to Verify how much secure the application is.
   In security testing, the tester focuses on Authentication and Authorization.

Authentication vs Authorization

Authentication : Users are valid or not

Authorization: Permission of Valid users

3. Recovery testing: Recovery testing is a type of testing that is used to verify the system is able to get back the data what it lost.

4. Compatibility Testing: Compatibility testing is a type of testing that is used to verifiy the system is working efficiently across different platforms such as browsers, devices, OS, network, and hardware.

5. Installation testing: Installation testing is a type of testing that is used to verify the installation process is working properly.

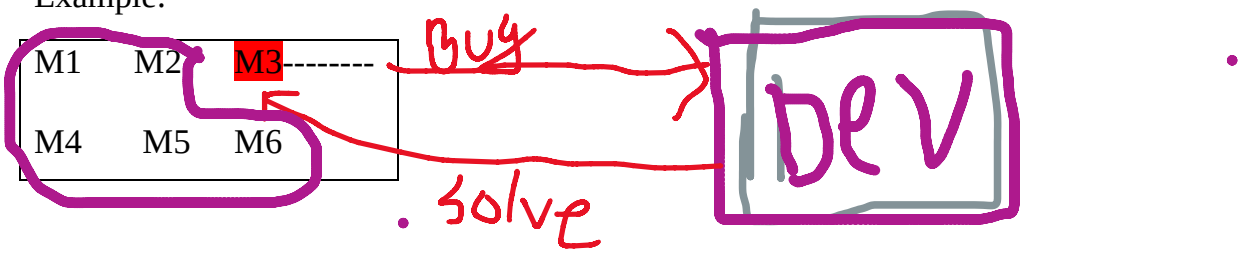
6. Sanity testing: Sanity testing is a type of software testing that is used to verify the small changes or new builds are working correctly.

7. Configuration testing:

Regression Testing vs Re-Testing

Regression testing: Regression testing is a type of software testing that is conducted on the modified build/changed build to ensure that there is no impacts on the existing functionality. And the product works fine with a new build that is regression testing.

Example:



Different types of Regression testing:

- Unit Regression testing
- Regional Regression testing
- Full Regression testing

Unit Regression testing: Testing only the changed/modified module to find bug that is Unit Regression testing.

Regional Regression testing: Testing the modified module along with the impacted module is called Regional regression testing.

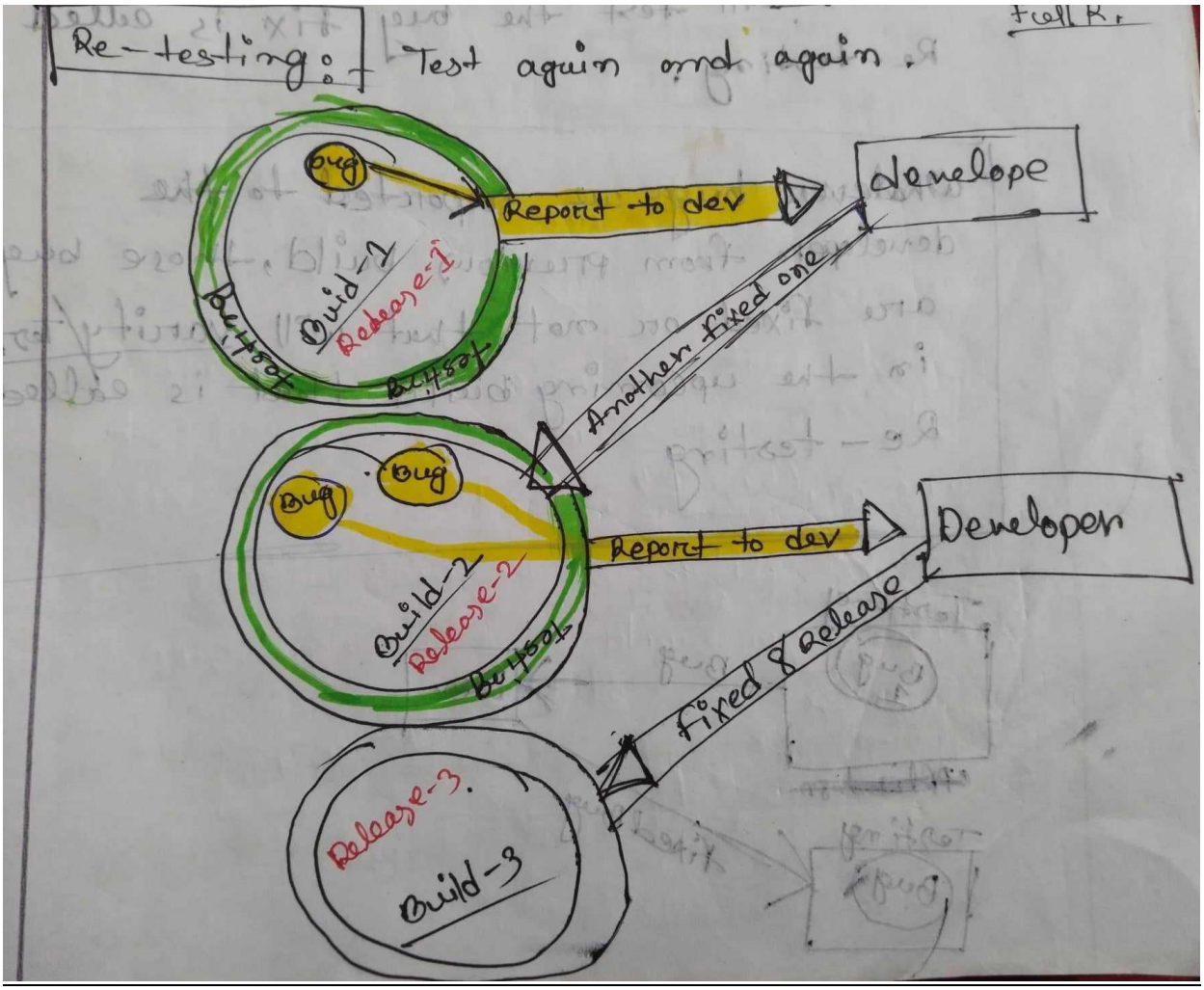
How to know the impacted area?

Ans: There will be a meeting conducted by the QA and Dev that meeting is called Impact Analysis. This impact analysis identifies the impacted area.

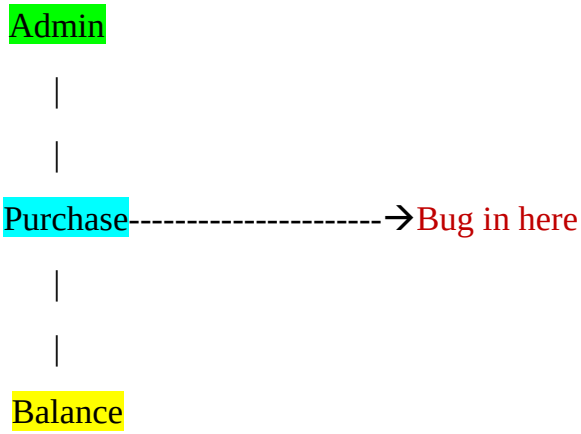
Full Regression testing: Testing the modified module along with the remaining module of the application is called full regression testing.

Re-testing: Re-testing means test again and again. Basically, Whatever bugs are reported from the previous build , those bugs are fixed or not that will verify in the upcoming build that is called Re-testing.

Example:



Example of Re-testing vs Regression testing



Explaination: Purchase module will be tested again & again that is Re-testing. On the other hand Purchase module along with Admin module and Balance module will be tested to ensure the impact that is called Regression testing.

Exploratory testing: Exploratory testing is a type of testing that is used to explore the application without having any documentation that is called Exploratory testing. Exploratory testing is done only by using previous experience not the documentation.

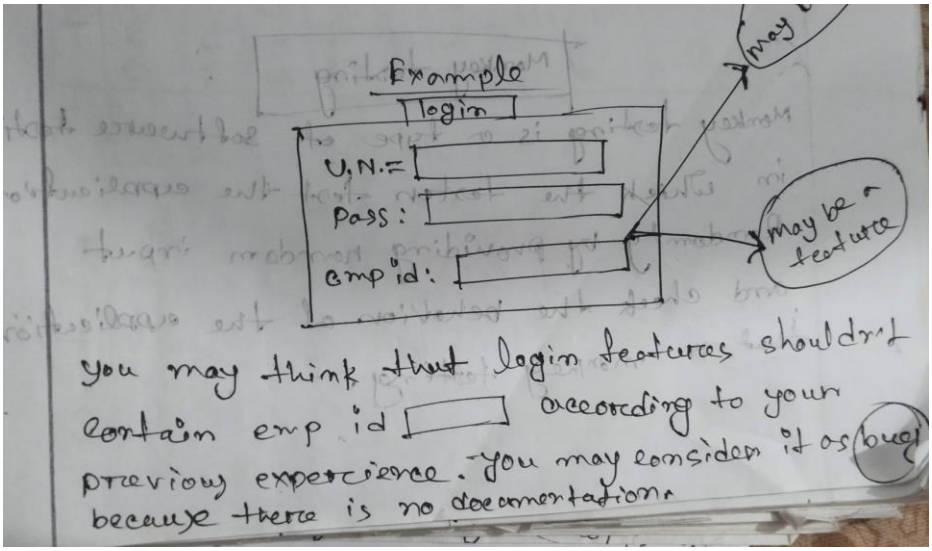
Note: We do exploratory testing when the application is ready but there is no documents

No documentation is available like (srs document, requirement document,design document) but your application is ready . Then you have to test the application by exploring the complete the application and understand it completely.

Drawbacks/Disadvantages of Exploratory Testing:

- Misunderstand of considering any bug as a feature or feature as a bug.
- If there is any bug you may not know about it.
- Time consuming.

Example :



Adhoc testing: Adhoc testing is type of testing in which tester test the application randomly without any documents , test cases, or test design that is called Adhoc testing.

Note: Randomly test the functionalities based upon the previous experience. This testing is usually an unplanned testing.

Monkey testing: Monkey testing is a type of software testing in which the tester tests the application randomly by providing random inputs and check the behavior of the application that is called monkey testing.

- Difference among Exploratory testing, Adhoc testing , and monkey testing

Exploratory testing	Adhoc testing	Monkey testing
No documentation.	No documentation	No documentation
Informal testing	Informal testing	Informal testing
Random testing	Random testing	Random testing
No plan	No plan	No Plan
Any Application	Gaming Application	Any Application

Smoke testing VS Sanity testing

Smoke testing: Smoke testing is a type of testing that determines whether the build/App is stable or not that is called smoke testing. It is also known as build verification testing.

What is build?

Ans: Build is nothing but software that contains a few numbers of features.

Smoke testing:

Smoke testing can be done by Dev or tester.

Smoke testing will be conducted on the new build to check the stability

How will we know the build is stable or not?

Ans: Installation of the software should be successful and basic navigation should work properly.

Sanity testing: Sanity testing is type of testing that is used to verify the small change/ modified build is working properly or not.

Positive testing Vs Negative testing

Positive testing: Testing the application with positive/ valid input is called positive testing.

Example:

Enter Only the Numbers

O123456789

Negative testing: Testing the software application with invalid/Negative input is called Negative testing.

Example :

Enter only the Numbers

acdefhgijklm

End to End testing: Testing the overall functionality which contains each and every component of the application sequentially.

STLC Model (software testing life cycle)

STLC Model: The software testing life cycle is a systematic approach to testing a software application and it ensures the high quality of software, and reliability, and meets the needs of end-user.

Phases of STLC : (RTTE-TT)

1. **Requirements Analysis :** Requirement Analysis is the first step of software testing life cycle. In this phase, QA team understand the requirement for project, if there is anything missing then QA team arrange a meeting with the stakeholder to understand the requirements.

2. **Test Planning** : Test planning is the most efficient phase in software testing life cycle. This phase starts only after the completion of requirements analysis phase. In this phase, the testing team manager calculates the i)estimated effort: identify the objective & strategy ii) cost : calculates the cost for project iii) schedule(time): identify the required time to complete the project.

3. **Test Case Development**: The test case development phase gets started once the test planning phase is completed. In this phase testing team notes down the detailed test cases and prepares the required test data for the testing.

4. **Environment Setup :** The testing team is not involved in this phase, Basically, environment set up means which software is suitable for this environment to be tested.

5. **Test Case Execution** : In this phase, the Testing team executed the test based on prepared test cases.

6. **Test case closure**: This is the last stage of STLC where the testing process is analyzed

What Test plan contain?

Ans: These iteams/contains should be in your test plan document.

- **Overview:** Small description
- **Scope:** Scope means what to test and what not to test.
- **Test strategy:** What kinds of strategy we are going to perform. Like (Automation testing or manual testing) (Regression Testing or Sanity testing)
- **Defect Reporting Procedure**: How we are going to report the bugs and what the exact process we follow.
- **Roles and Responsibilities:** In a testing team there are multiple people will be assigned for specific roles like (test engineer,QA lead , project manager)
- **Test schedule**: When we have to conduct what kind of testing.
- **Pricing:** How much price it will be taken .
- **Entry and exit criteria:** when you have to starts and stop the testing .
- **Tools:** what kind of tools we are going to use for testing.
- **Test delivery:**
- **Risks:**
- **Approvals**

Use case , Test Scenario and Test case:

Use case: Use case describe the requirements graphically ,(Describe the requirement).

Note : Use case is a data flow diagram which contain requirements

Use case contain :

- ✓ **Actor:** User
- ✓ **Action:** Process
- ✓ **Goal/Outcome:** Result

Test Scenario: Test scenario is a possible area to be tested.that's mean a possible are to be tested (what to test).

Test Case: Test case is a step by step action to be performed to validate the functionality of the application (How to test)

Difference between Use case and Test case

Use case : The use case describes the functional requirements prepared by Business Analyst.

Test case: The test case describes the step-by-step procedure prepared by Test Engineer.

Diagram:

(1) Use case --------Requirement
(2) Test Scenario-----What to test
(3) Test Case--------How to test

Difference between Test scenario and Test Case

We will get test Scenario from use case , Test scenario is "what to test"

Test case is "How to test"

Example:

Checking the functionality of Login Button

TC-1: Click the button without entering user name and password.

TC-2: Click the button only entering User name

TC-3: Click the button while entering wrong User name and wrong password

Here, Login ---Test Scenario And TC-1 TC-2 and TC-3-----Test case

Test Suite: A Test Suite is a group of test cases that belongs to same. Category like (Sanity test suite , Regression test suite)

Test Case: Test case is a step-by-step action that is used to validate functionality of the software application.

Test Case contents:

1) Test ID
2) Test Case Title
3) Description
4) Pre-condition
5) Priority
6) Requirement ID
7) Step Action
8) Expected Result
9) Actual Result
10) Test Data

Requirement Traceability Matrix (RTM)

RTM describe the mapping of the requirements with the test cases.

The Purpose of RTM: The purpose of RTM is to see that all the test cases are covered so that no functionality should miss while doing software testing.

Reg No	Reg description	Test case ID	Status
123	Login the App		
345	Ticket Creation		
456	Search Ticket		

What is RTM: Mapping between requirement IDs and Test case ID is called RTM.

Test Environment: Test environment is a platform specially build for test case execution on the software product. It is also known as Test Bed.

Test Criteria: Test case, Test data, and test plan Before starting the testing , you should have the test case ready and approved, test plan is required for maintaining the timelines and also test data.

Activities:

i. Test cases are executed based on the test plan
ii. Documentation of the test result

Bug/Defect Reporting Tools

- Clear Quest
- DevTrack
- Jira
- Quality center

Bug Jilla

Difference between Test management Tools and Defect Tracking Tools

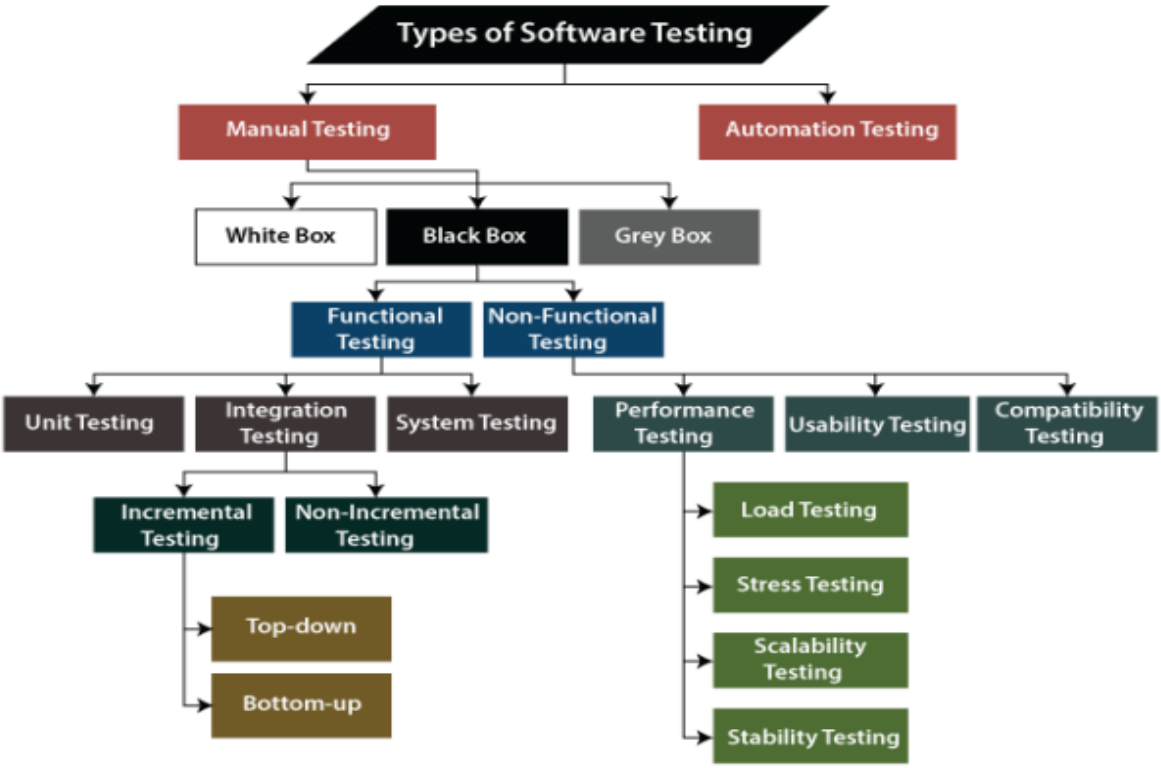
Test management tools : In test management tools we can define requirements, we can write test scenario, we can write test cases, upgrade the execution status and report the defect.

Note: Not only defect reporting , we can also do other test management activities.

Tools: Jira, QualityCenter

Defect tracking Tools: It can only tracking the defects

Tools: ClearQuest, DevTrack, Bugzilla.



Software Testing : Software testing is a process of executing a program to find errors.

1. **Manual Testing:** Manual testing is a software testing process in which test cases are executed manually without using any automated too

2. **Automation Testing :** Automation testing is a software testing process in which test cases are executed manually with using any automated too
3.

Black box Testing: Black box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation details of the software but focuses on the functionalities that is called Black box testing.

Black box test design techniques,

Test Design Techniques: Test design techniques are used to prepare test data.

Purpose of test design technique: The purpose of test design technique is to reduce the test data and increase the coverage.

Five types of test design technique:

I. Equivalence partitioning
II. Boundary Value Analysis (BVA)
III. Decision Table-based testing
IV. State transition
V. Error guessing.

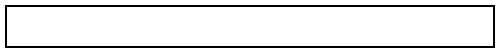
Equivalence Partitioning: Equivalence partitioning is a type of Black -Box testing technique where input data units are divided into valid and invalid partitions. It is also Known as (Ecp) Equivalence Class Partitioning.

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

Note: In ECP we will focus on the value Partition.

Example: Allows digits from 1 to 500.

Enter a number:



Normal test data:

1

2

3

4

5

.

.

,

,

500

Divided Values into Equivalence classes:

-100 to 0-→ -50 (invalid) (pick only one value among this)

1 to 100 -→ 30 (Valid) (pick only one value among this)

101 to 200--→ 160 (Valid)) (pick only one value among this)

201 to 300 -→ 250 (Valid) (pick only one value among this)

301 to 400 -→ 320 (Valid) (pick only one value among this)

401 to 500→ 450 (Valid) (pick only one value among this)

501 to 600→ 502 (invalid) (pick only one value among this)

Test Data using ECP:

-50	30	160	250	320	450	502

Note: By using ECP techniques , we can minimize the test data and test Coverage.

One more example: Allows only Alphabets

Name

Divided values into Equivalence classes

A to Z -→ D (valid)

a to z -→ f (valid)

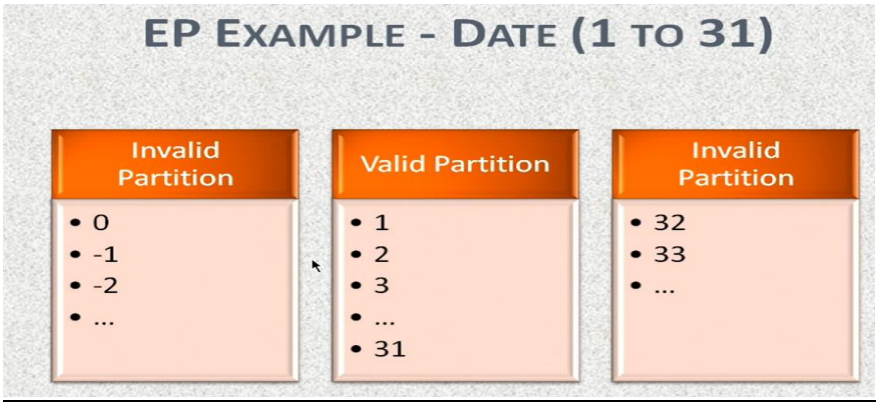
Special characters: @#$%&*% -→ $ (invaid)

Spaces: xy z (invaid)

Numbers: 1 to 500 → 333 (invalid)

Test Data

D	f	@#$%&*%	xy z	1 to 500	



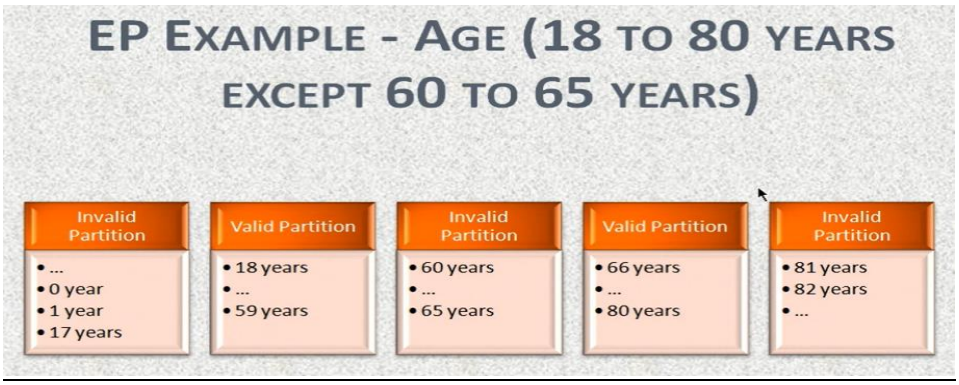


Fig : Example of equivalence Partitioning

Note: Why use equivalence partition?
Ans : Equivalence Partitioning allows you to find more defects because you are testing each and every partition.

I) Boundary Value Analysis: Boundary value analysis is a black box testing technique that involves analyzing with the parameter of the boundaries.

In BVA we will focus on Boundary of the value . it just test the Boundaries.

Allows digit from 18--35

Enter a Age:

-(Min-1)-------Min-------(Min+1)----------(Max-1)-------(Max)----------(Max+1)

-(18-1)---------(18)------(18+!)-------------------------(35-1)----------(35)-------------(35+1)

Test Data:

Min = 18	(valid)	Max =35	(valid)
Min-1 = 17	(valid)	Max-1 = 34	(valid)
Min+1=19	(Invalid)	Max+1 =36	(Invalid)

Need to Remember to identify the boundary values:

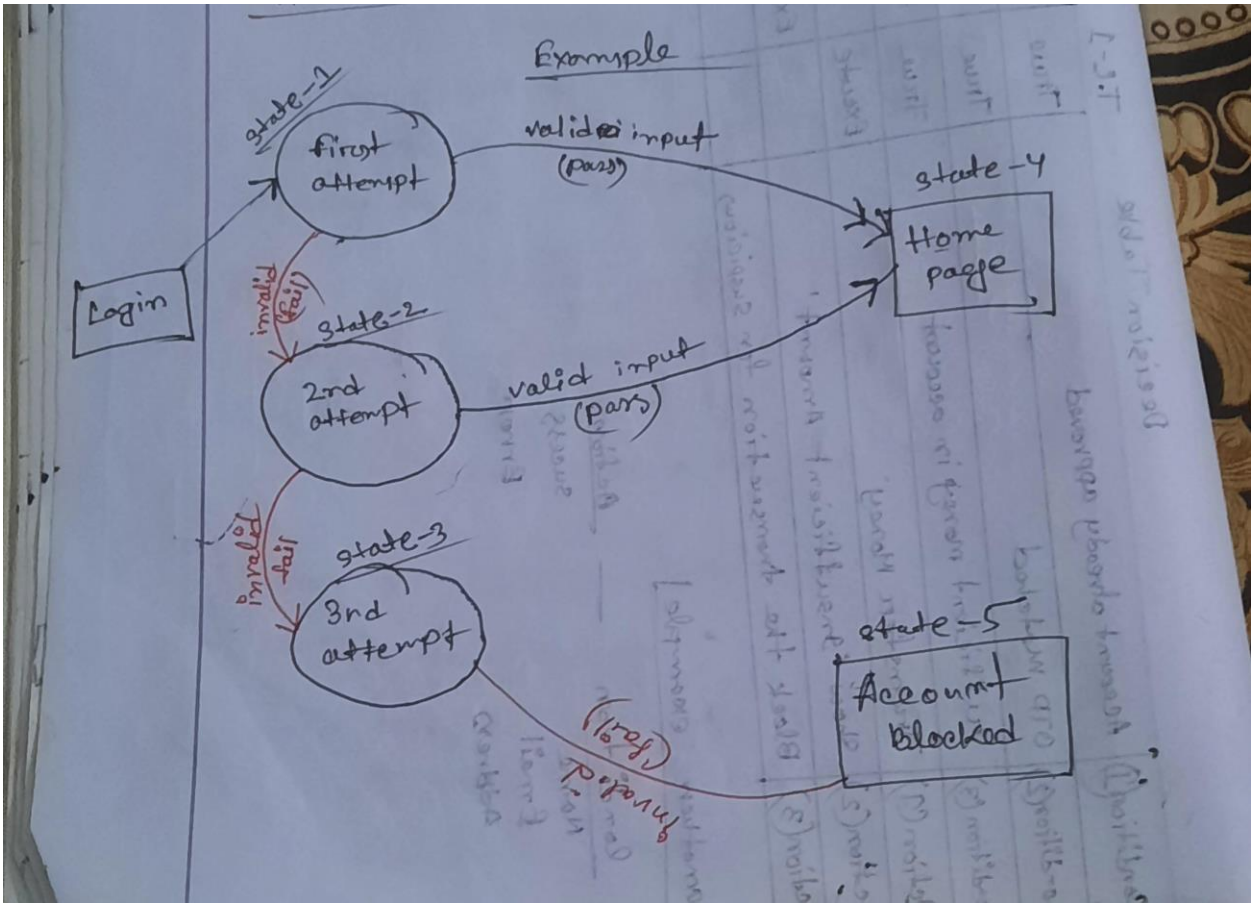
i. Minimum value
ii. Just above the minimum value
iii. Normal value
iv. Just below the Maximum value
v. Maximum value

Example :

BOUNDARY VALUE ANALYSIS		
Invalid (min -1)	Valid (min, +min, -max, max)	Invalid (max +1)
17	18, 19, 55, 56	57

.

II) State Transition testing: State transition testing is a black box testing technique that is performed to check the change in the state of the application under varying input. It observes the behavior of the system using different input conditions



State	Login	Correct Password	Incorrect Password
S-1	1st Attempt	S-4	S-2
S-2	2nd Attempt	S-4	S-3
S-3	3rd Attempt	S-4	S-5
S-4	Home Page		
S-5	Account Locked		

III) Decision Table : A decision table is a software testing technique that is used to test system behavior for different input combinations. All these input and output combinations are captured in a tabular form.

Example : Take an example of transferring money in online to an account that is already added and approved.

- Here the Condition to transfer money:
  ---Account already approved .
  ----OTP matched .
  ----Sufficient money in the account
- And the actions performed are:
  ---Transfer money
  ---Show a massage as insufficient amount
  --Block the transaction.

Example: (TC-Test Case)

		TC-1	TC-2	TC-3	TC-4	TC-5
Condition-1	Account already approved	True	True	True	True	False
Condition-2	OTP matched	True	True	False	False	X
Condition-3	Sufficient money in account	True	False	True	False	X
Action-1	Transfer money	Execute				
Action-2	Show “Insufficient Amount”		Execute			
Action-3	Block the transaction			Execute	Execute	

Another example:

Condition- Action

Name ---------- Success

Email ----------Error

Address

White box Testing: White box testing is a type of software testing in which the tester involves with internal structure, design, and coding . The tester has access to the source code and uses this source code in the design. It is called white box testing. It is also known as glass box testing or clear box testing, structural testing, transparent testing, or open box testing.

White box testing is known by these names (glass box testing/ clear box testing/ structural testing / transparent testing /open box testing)

White box testing techniques:

1. Statement coverage: All the executable statements of the source codes are executed at least once. It is used for the calculation of the number of statements that have been used.

- $test\ coverage = \frac{lines\ of\ code\ covered\ by\ tests}{total\ lines\ of\ code} * 100$

Scenario- 1

```
Prints (int a, int b) {
int result = a+ b;
If (result> 0)
    Print ("Positive", result)
Else
    Print ("Negative", result)
}
```

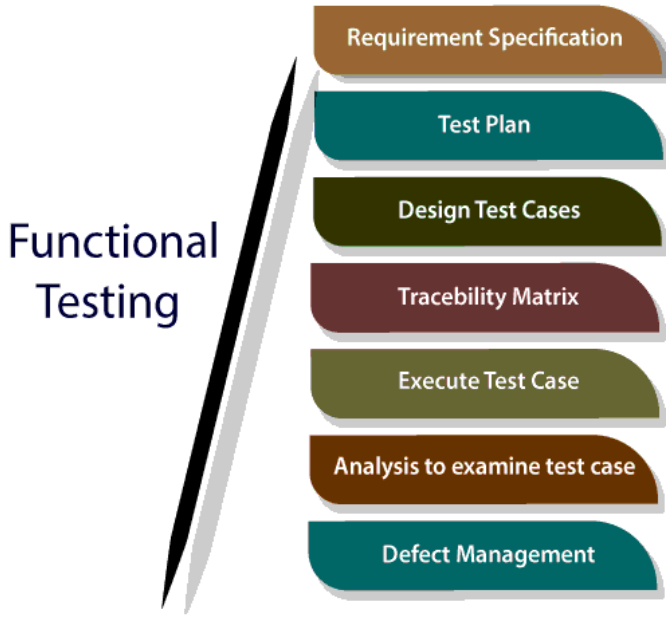
Scenario- 2

```
Prints (int a, int b) {
int result = a+ b;
If (result> 0)
    Print ("Positive", result)
Else
    Print ("Negative", result)
}
```

2. Branch coverage.
3. Condition coverage.
4. Path coverage.

Grey Box testing; Greybox testing is a type of software testing that tests the software application with partial knowledge of the internal structure. It is a combination of black box and white box testing

Grey box testing techniques:

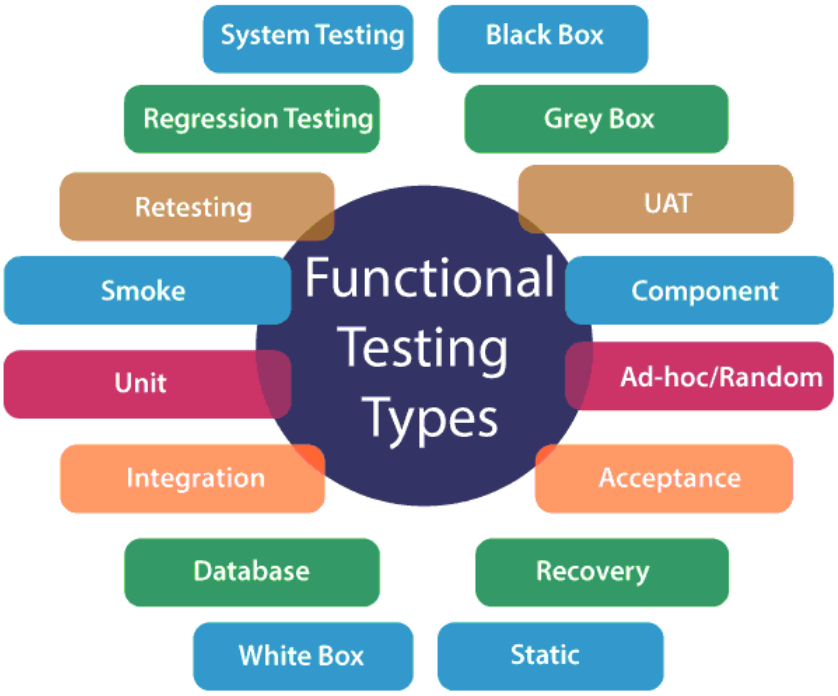


What to test in functional testing?

The main objective of functional testing is to check the functionality of the software system.

- Basic useability: It checks whether a user can use/navigate freely without any difficulty.
- Accessibility: Functional testing tests the accessibility of function.
- Mainline Function: It focuses on testing the main feature of the function.
- Error condition: Functional testing is used to check the error condition, whether the error is displayed or not.

Types of Functional Testing:



What are the functional testing tools ?

Tools	Features/ Characteristics
Sahi	◦ It is an open-source and automation testing tool, released under Apache License open source license, used for testing of the web application. ◦ Sahi is written in Java and JavaScript and considered for most of the testing techniques. ◦ It runs as a proxy server; it is browser-independent.
SoapUI	◦ It is an open-source functional testing tool, used for web application testing. ◦ It is simple and easy to design. ◦ It supports multiple environments, i.e., at any instance, the target environment may be set up.
Watir	◦ Watir, is an abbreviated form of web application testing in ruby, is an open-source tool for automating web browser./li> ◦ It uses a ruby scripting language, which is concise and easy to use./li> ◦ Watir supports multiple browsers on various platform.
Selenium	◦ The open-source tool, used for functional testing on both web application and applications of the desktop.
	◦ It automates browsers and web application for testing purpose. ◦ It gives the flexibility to customize the automated test case ◦ Provides the advantage of writing test scripts, as per the requirements, using web driver.

	◦ It automates browsers and web application for testing purpose. ◦ It gives the flexibility to customize the automated test case ◦ Provides the advantage of writing test scripts, as per the requirements, using web driver.
Canoo WebTest	◦ An open-source tool for performing functional testing of the web application. ◦ Platform independent ◦ Easy and fast ◦ Easy to extend to meet growing and incoming requirements.
Cucumber	◦ Cucumber is an open-source testing tool written in Ruby language. This tool works best for test-driven development. It is used to test many other languages like java, c#, and python. Cucumber for testing using some programming.

SDLC	STLC
1. SDLC is mainly related to software development.	1. STLC is mainly related to software testing.
2. SDLC helps in developing good quality software.	2. STLC helps in making the defects free software
3. SDLC phases are completed before the STLC phases.	3. STLC phases are performed after the completion of SDLC phases.
4. SDLC process needs a greater number of developers to complete the development process	4. The STLC process needs a smaller number of testers to complete the testing process.
5. Project Managers, Business analysts, and Developers, are involved in SDLC.	5. Quality assurance and Testers teams are involved in STLC.

Seven Principle of Software Testing

There are seven principles of Software Testing

1) Testing shows the presence of defects:

"Absence of proof is not the proof of absence ".

It doesn't mean that there aren't any bugs in the product. It just means that there may be bugs, but you didn't find them.

2) **Exhaustive testing is not possible:** The truth is that it's impossible to test all the modules throughout the actual testing process. To ensure that you are testing the essential functions, you have to analyze the risks involved and plan your tests properly.

3) **Early Testing:** Early testing is the key to identifying any defects in the requirements or design phase as soon as possible. It's much easier and less expensive to fix bugs in the early stages of testing than at the end of the software.

4) **Defect Clustering:** **" Approximately 80% of the problems are found in 20% of the modules."** It's called Pareto Principle. This means the small number of modules contain most of the defects detected.

5) **Pesticide paradox**: If you continuously run the same tests in different modules, then they'll fail to find new defects, even though they'll probably confirm the software is working.

6) **Testing is context dependent**: Software testing is all about the context, which means that the same strategy will not fit in every scenario. The types of testing depends on the context.

   For example: The testing of an iOS application is different from the testing of an e-commerce website.

7) **Absence-of-errors fallacy:** If the software or system does not fulfill users' wishes then it does not matter how many defects are found and fixed. It is still unusable. So in this sense, it is irrelevant how error-free your system is;