Session – 1

software :- A software is a collection of computer programs that helps to perform a task.

Type of software :-

1. System software :- device drivers, operating systems, servers, Utilities, etc.
2. Programming software:- compilers, debuggers, interpreters, etc
3. Application software:- Web applications, mobile apps, desktop application,

What is software Testing :-

software testing is a part of software development process.

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

The objective of testing is to release quality product to the client

Software Quality :- (quality means it is a customer justification we can measure it by below parameters)

* + bug-free
  + delivered on time
  + within budget
  + meets requirements and /or expectations.
  + maintainable.

Project VS Product :-

* If software application is developed for specific customer based on the requirement then it is called project.
* If software application is developed for multiple customers based on market requirements then it called product.

Why do we need testing :-

The objective of testing is to release quality product to the client and ensure the software should be bug free and meet the customer requirement.

Error , bug/defect, & Failure :-

Error – it is a human mistake while writing a program developer will do lots off mistake

bug/defect = while testing the software we are checking the functionality of our application weather it is working according to the customer requirement or not if something is not working according to customer requirement which is come under the bug/defect

Failure = after releasing the product to the customer and customer is using the software where he is found some bugs , mismatch , something is not working which is come under the Failure.

Why The software has bug :-

* Miscommunication or no communication
* software complexity
* programming errors
* changing requirements
* lack of skilled testers

Session – 2

SDLC (software development life Cycle )

software development life cycle is a process used by software industry to design, develop and test software’s.

P- People

P- Process

P- Product

- these are the three pillers of any company

Requirement analysis

Design

Maintenance

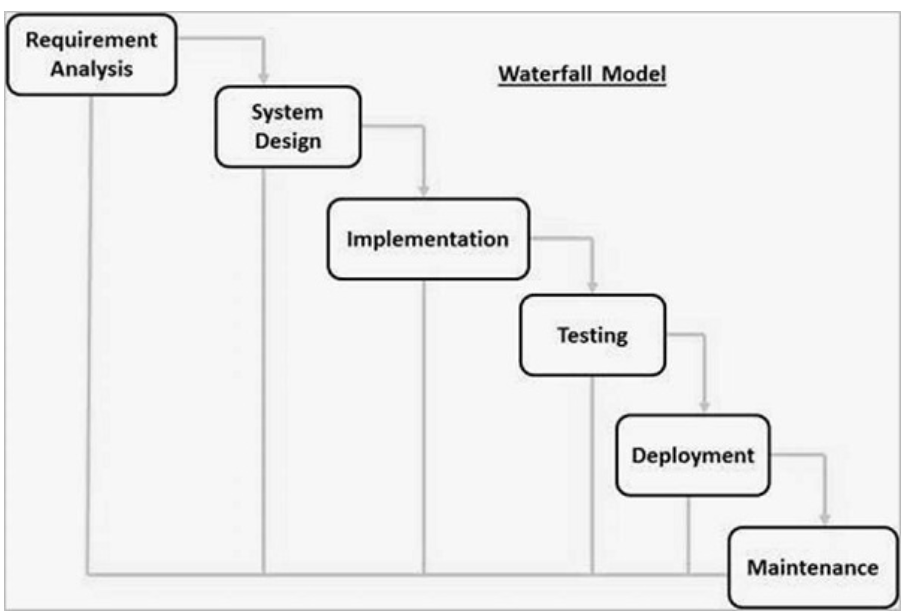
Testing

Development

Following are the most important and popular SDLC models followed in the industry −

* Waterfall Model
* Iterative Model
* Spiral Model
* V-Model
* Big Bang Model

Waterfall model :-



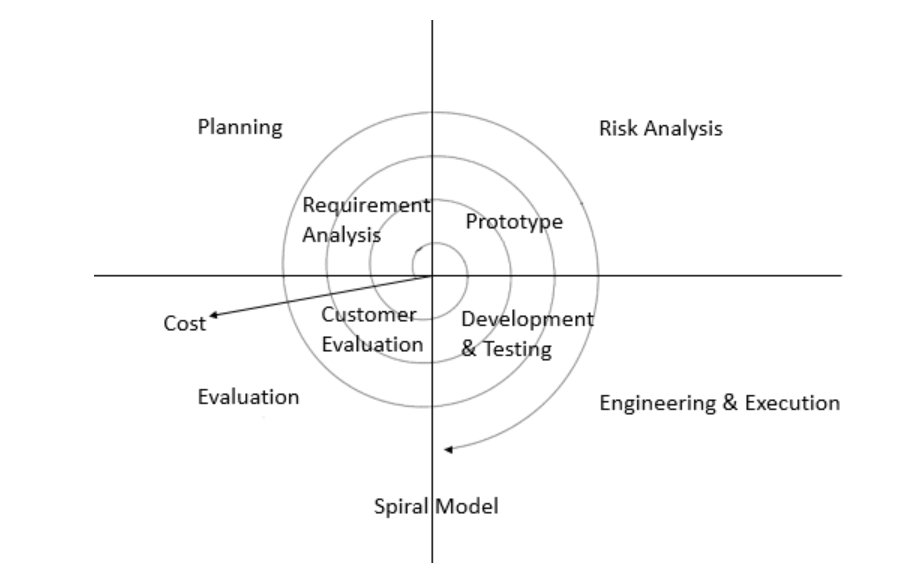
Advantage-

* quality of the product will be good
* since requirement changes are not allowed, chances of finding bugs will be less.
* initial investment is less since the tester are hired at the later stages.
* preferred for small projects where requirements are feezed.

Disadvantage-

* Requirement changes are not allowed.
* if there is defect in requirement that will be continue in later phase.
* total investment is more because time taking for rework on defect is time consuming which leads to high investment.
* testing will start only after coding.

Spiral model :-



* spiral model is iterative model.
* spiral model overcome drawbacks of waterfall model.
* we follow spiral model whenever there is dependency on the modules.
* in every cycle new software will be released to customer.
* software will be released in multiple versions. so it is also called version control model.

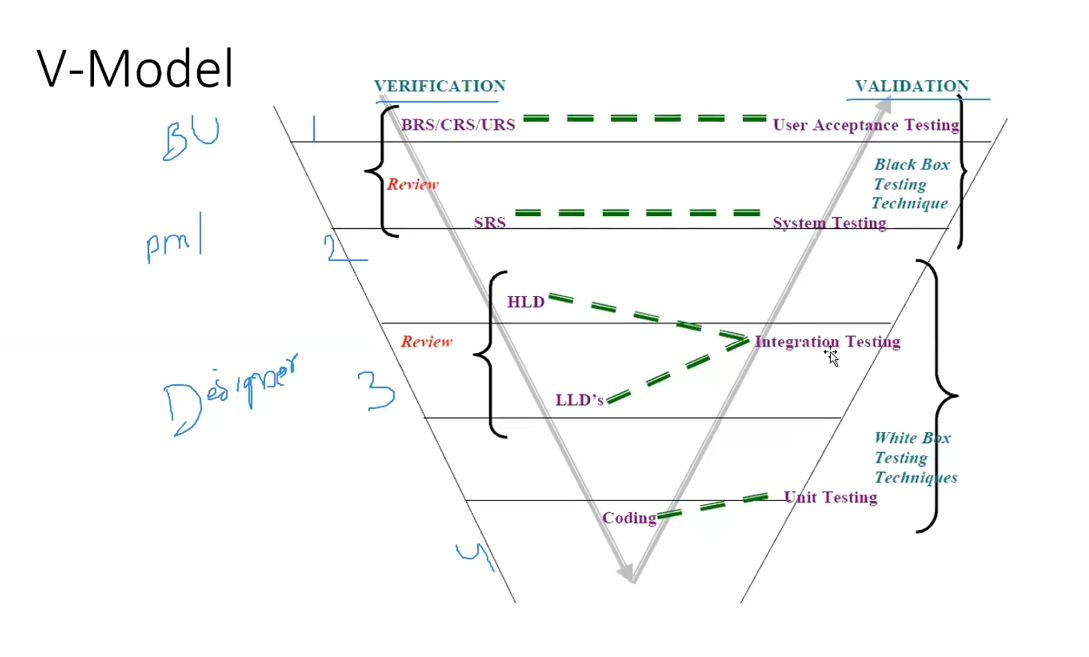
Advantage –

* Testing is done in every cycle, before going to the next cycle.
* customer will get to use the software for every module.
* requirement changes are allowed after every cycle before going to the next cycle.

Disadvatage –

* Requirement changes are not allowed in between cycle.
* every cycle of spiral model looks like waterfall model.
* there is no testing in requirement

V –model/ VV model -



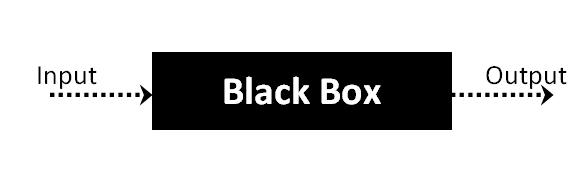
Static testing :- Testing the project related documents is called as static testing.

In the form of – review, walkthrough, inspection

White Box Testing : whatever the testing we conduct on the code or sub modules or main modules which is called as White box testing here unit testing and integration testing comes under in the white box testing techniques this testing will be developer himself.

esme code test krte hai application nhi .

**Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.



black box testing m hume nhi pta hota ander kya hai(code) bs functionality test krte hai according to customer requirement ki software shi work kr rha hai ya nhi esme system testing or user acceptance testing aati hai

Dynamic Testing :- Testing the actual software, we are using technique like -

* Unit testing.
* Integration Testing
* System Testing
* UAT Testing

What is Verification and Validation :-

Verification :- verification checks whether we are building the right product.

* focus on documentation
* verification typically involves.
* reviews
* walkthroughs
* inspections

Validation :- checks whether we are building the product right.

* take place after verifications are completed.
* focus on software
* validation typically involves actual testing.
* Unit testing , integration, system testing, UAT testing

Advantage :- Testing involved in each and every phase.

Disadvantage :-

* documentation is more.
* initial investment is more.

Session – 3

Static Testing Technique :

1. review
2. walkthrough
3. inspection

Dynamic Testing Technique:

1. unit testing
2. integration testing
3. system testing
4. UAT

What is Review : -

* Conducts on documents to ensure correctness and completeness.

Different type of Review :-

* Requirement Review
* Design Reviews
* Code Reviews
* Test plan reviews
* Test cases reviews etc.

Walkthrough :-

* it is informal review.
* author reads the documents or code and discuss with peers.
* it’s not pre-planned and can be done whenever required.
* also walkthrough does have minutes of the meets.

Inspection :-

* it’s 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.

QA vs QC :-

QA is process related.

QC is the actual testing of the software.

QA focuses on building in quality.

QC focuses on testing for quality

QA is preventing defects.

QC is detecting defects.

QA is process oriented.

QC is product oriented.

QA is for entire life cycle

QC for testing part in SDLC

P- people =QC(testers its only come under in Testing)

p- process = QA (come under higher management ye pure project m rhte hai)

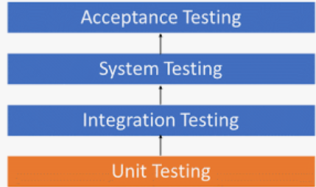
p- product

SE – Software Engineer – they write the code for developed a software

QE – Quality Engineering – they also write the code for testing , ye automation tester hote hai

Levels of Testing :

1. Unit testing
2. integration testing
3. system testing
4. user acceptance testing (UAT)



Unit Testing :-

* A unit is a single component or module of a software.
* unit testing conducts on a single program or single module.
* unit testing is white box testing technique
* unit testing is conduct by the developers

Unit testing techniques :

* basis 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 between multiple modules.
* Integration testing is white box testing.

Types of integration testing :

1. Incremental integration testing
2. non-incremental integration testing

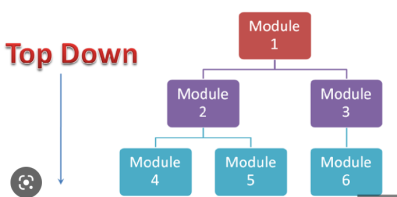
Incremental integration testing :- incrementally adding the modules and testing the data flow between the modules.

its have two approaches :-

1. Top Down
2. Bottom Up

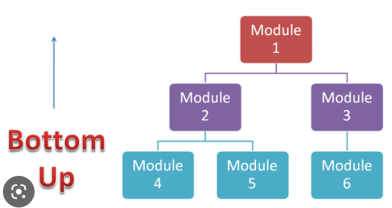
**Top Down** :- incrementally adding the modules and testing the data flow between the modules. and ensure the module added is the child of previous module.

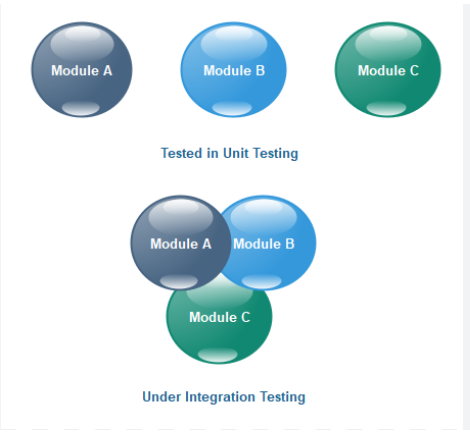
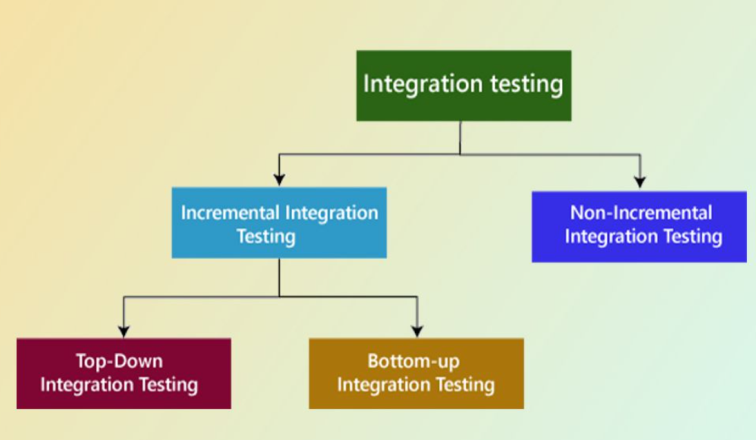
emp – jaise gmail m composed parent hai or sent idem child hai qki pahle send krenge jabhi to send item m mail aayegi



**Bottom Up :-** incrementally adding the modules and testing the data flow between the modules. and ensure the module added is the parent of previous module.

exp – esme pahle sent item m dekhenge agr item hai to composed ki gyi hai

****

Sandwich/Hybrid Approach : combination of Top – down & Bottom up approach is called sandwich approach

non-incremental integration testing :- Adding all the modules in a single shot and test the data flow between modules.

drawback-

* we might miss data flow between some of the modules.
* if you find any defect we cant understand the root cause of defect.

System Testing :-

* Testing over all functionality of the application with respective client requirements.
* It is a black box testing technique.
* This testing is conducted by testing teams.
* After completion of component and integration level testing’s we start System testing.
* Before conducting focusses on below aspect

1. User Interface Testing (GUI)
2. Functional Testing
3. Non-Functional Testing (like performance testing,security)
4. Usability Testing

Session – 4

What is GUI Testing :-

* **Graphical User-Interface Testing** or **GUI Testing** is a process of **testing** the user interface of an application.
* A **graphical user interfae**  includes, all the elements such as menus, checkbox, buttons, colors, fonts, sizes, icons, content, and images.
* it is mainly focus on UI or front-end part

**GUI Testing Checklist :-**

* testing the size, position, width, height of the element
* testing of the error messages that are getting displayed.
* testing the different sections of the screen.
* testing of the font whether it is readable or not.
* testing of the screen in different resolutions with the help of zooming and zooming out.
* testing the alignment of the texts and other elements like icons, button, etc. are a proper place or not.
* testing the colors of the fonts.
* testing whether the image has good clarity or not.
* testing the alignment of the image.
* testing of the spelling.
* the user must not get frustrated while using the system interface.
* testing whether the interface is attractive or not.
* testing of the scrollbars according to the size of the page if any.
* testing of the disabled field if any.
* testing of the size of the images.
* testing of the headings whether it is properly aligned or not.
* testing of the color of the hyperlink.
* testing UI elements like button, textbox, text area, check box, radio buttons, drop downs, links etc.

Usability Testing :-

* during this testing validates application provided context sensitive help or not the user.
* checks how easily the end users are able to understand and operate the application is called usability testing.

Functional Testing :-

* functionality is nothing but behaviour of application.
* functional testing talks about how your feature should work.
* **object properties testing** (check the properties of objects present on the application, exp – enable and disable, visible, focus)
* **database testing/backed testing**(mainly focus on the DML operation working fine or not, main operations insert, update, delete, select)
* **error handling** (tester verify the error messages while performing incorrect actions on the application, error messages should be readable,user understandable/simple language)
* **calculations/manipulations testing**(tester should verify the calculations)
* **links existence & links execution**(where exactly the links are placed –**links existence**, links are navigating to proper page or not - **links execution , inter links-** which link navigate on the same page in the differ section

**exter links–**when I click on the link which will navigate to some other page

**brokens links –** when I click on link but they will not navigate anywhere else)

* **cookies &sessions** (cookies – temporary files created by browser while browsing the pages through internet

Sessions- are time slots created by the server. session will be expired after some time if you are idle for some time)

**Greybox testing kya hai :-** jisme white box or black box dono ka use ho usko khte hai grey box testing example – database testing esme ui part m data insert krke dekhte hai or backed m verify krte hai data shi se insert , delete or update hua ya nhi esme don technique ka use hua hai.

Non-Functional Testing :-

* once the application functionality is stable then we do non-functional testing.
* focus on performance, load it can take and security etc.

1. **performance testing** - speed of the application

these three testing comes under the performance testing-

load testing – **gradually** increasing the load on the application slowly then check the speed of the application, tools – load runner jmeter etc

stress testing – **suddenly** we increase/decrease the load on the application and check the speed of the application

volume testing- check how much data is able to handle by the application

1. Security testing :- we are verifying how secure our application and few things we are verify-

authentication – users are valid or not

autherization/access control—permissions of the valid user.

1. Recovery testing :- check the system change to abnormal to normal.
2. compatibility testing :-

former compatibility

backward compatibility

hardware compatibility (configuration testing)

1. Installation testing :- -check screens are clear to understand.

-screens navigation

-simple or not

-un-installation

1. sanitation /garbage testing :- if any application provides extra features/functionality then we consider them as a bug.

|  |  |
| --- | --- |
| Function Testing | Non-functional Testing |
| * Validates functionality of software * functionality describes what software does. * concentrates on user requirement. * functional testing take place before non-functional testing. | * verify the performance, security,reliability of the software. * non-functionality describes how software works. * concentrates on user expectation. * non-functional testing performed after finishing functional testing. |

Session – 5

Software Testing Terminology :-

Regression Testing :-

* Testing conducts on modified build to make sure there will not be impact on existing functionality because of changes like adding/deleting/modifying features.
* Unit Testing –

Testing only the changes/modifications done by the developer.

* Regional Regression Testing –

- Testing the modified module along with the impacted modules

- impact analysis meeting conducts to identify impacted modules with QA & Dev.

* Full Regression –

-Testing the main feature & remaining part of the application.

-Ex: Dev has done changes in many modules, instead of identifying impacted modules, we

perform on round of the regression.

Re-Testing :-

* Whenever the developer fixed a bug, tester will test the bug fix is called Re-Testing.
* Tester close the bug if it worked otherwise re-open and send to developer.
* To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build.
* Example

-Build 1.0 was released. Test team found some defects (defect id 1.0.1, 1.0.2) and posted.

-Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

Example: Re-Testing Vs Regression Testing

* An application under Test has three modules namely **Admin**, **Purchase** and **Finance**.
* Finance module depends on Purchase module.
* if a tester found a bug on purchase module and posted. Once the bug is fixed, the tester needs to do **Retesting** to verify whether the bug related to the purchase is fixed or not and also tester needs to do **Regression testing** to test the finance module which depends on the purchase module.

Admin

Finance

Purchase

Smoke Testing : -

Smoke testing, also known as “Build Verification Testing” or “Build Acceptance Testing,” is **a type of software testing that is typically performed at the beginning of the development process to ensure that the most critical functions of a software application are working correctly**.

Sanity Testing – **it is perform on the new build to check new added functionality and bug working correctly or not**

by chat GPT definition =

Sanity testing ek aisa software testing hai jo kisi bhi software ya system ke minor changes ya bug fixes ke baad uske proper functioning ko evaluate karne ke liye kiya jata hai.

Sanity testing ka main objective hai ki verify karna ki software ke critical functionalities par kisi prakar ka asar nahi hua hai uske code ke changes ke karan. Ye testing regression testing ke baad ki jati hai aur usually application ya system ke ek chhote portion par perform ki jati hai.

Sanity testing ek high-level testing activity hai jo newly implemented changes ne software mein koi major issues ya defects to nahi introduce kiye hai wo identify karne mein help karta hai. Agar sanity testing ke dauran koi critical issues identify hote hai, to development team uss issue ko jaldi se fix kar sakti hai aur software ko end-users ke liye release kar sakti hai.

Overall, sanity testing ek quick and efficient way hai software ke minor changes ya bug fixes ke baad verify karne ke liye ki software ya system proper functioning kar raha hai ya nahi.

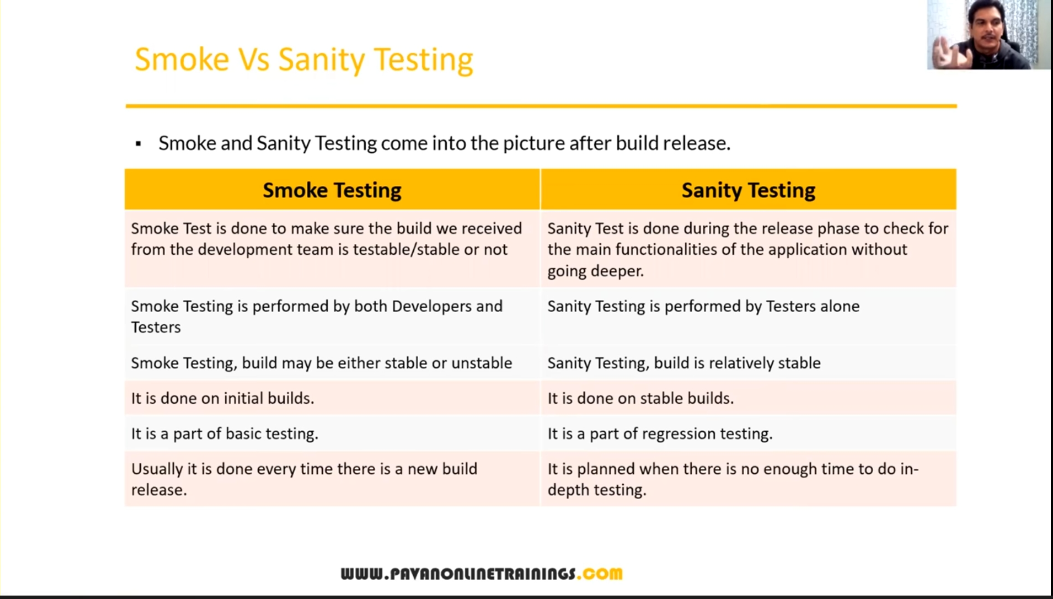
smoke testing kya hai

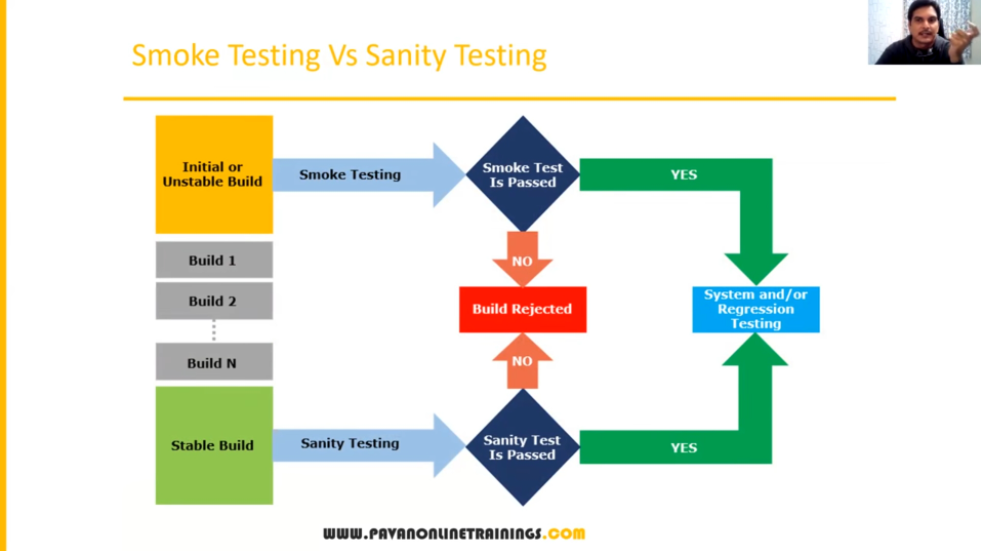
Smoke testing ek aisa software testing technique hai jo software ke basic functionalities ka quick test karne ke liye kiya jata hai, aur ye testing software development cycle ke early stage mein kiya jata hai. Smoke testing ko "Build Verification Testing" (BVT) ya "Confidence Testing" ke naam se bhi jaana jata hai.

Smoke testing ka main purpose hai ki verify karna ki software ke critical functionalities sahi se work kar rahe hai ya nahi. Is testing technique mein, software ke basic functionalities ko test kiya jata hai jaise ki login, homepage, navigation, search, etc. Smoke testing mein detailed testing nahi kiya jata hai, balki sirf software ke basic functionalities ka test kiya jata hai.

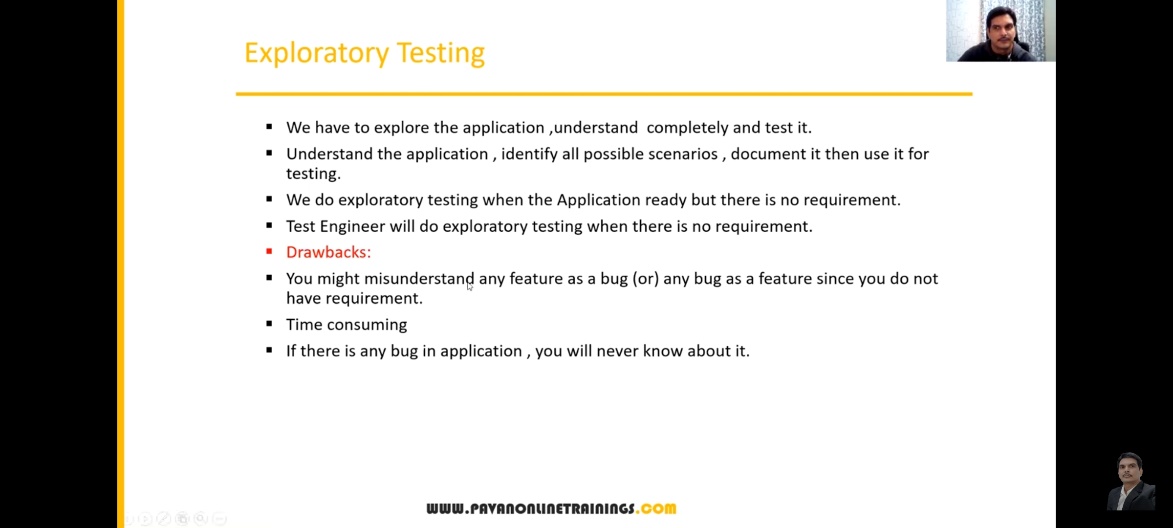
Smoke testing kiya jata hai before detailed testing ya regression testing ke liye, taki pata chale ki kya software ke major issues hai aur detailed testing ke liye software ready hai ya nahi. Agar smoke testing mein koi major issue identify hota hai toh development team usse fix karne ke baad smoke testing ko dobara se perform karte hai.

Overall, Smoke testing ek aasan aur fast way hai ki software ke critical functionalities sahi se work kar rahe hai ya nahi, aur iske liye detailed testing nahi kiya jata hai. Ye software development cycle ke early stage mein kiya jata hai aur software ke build verification aur confidence ko test karne mein help karta hai.





Exploratory Testing :-



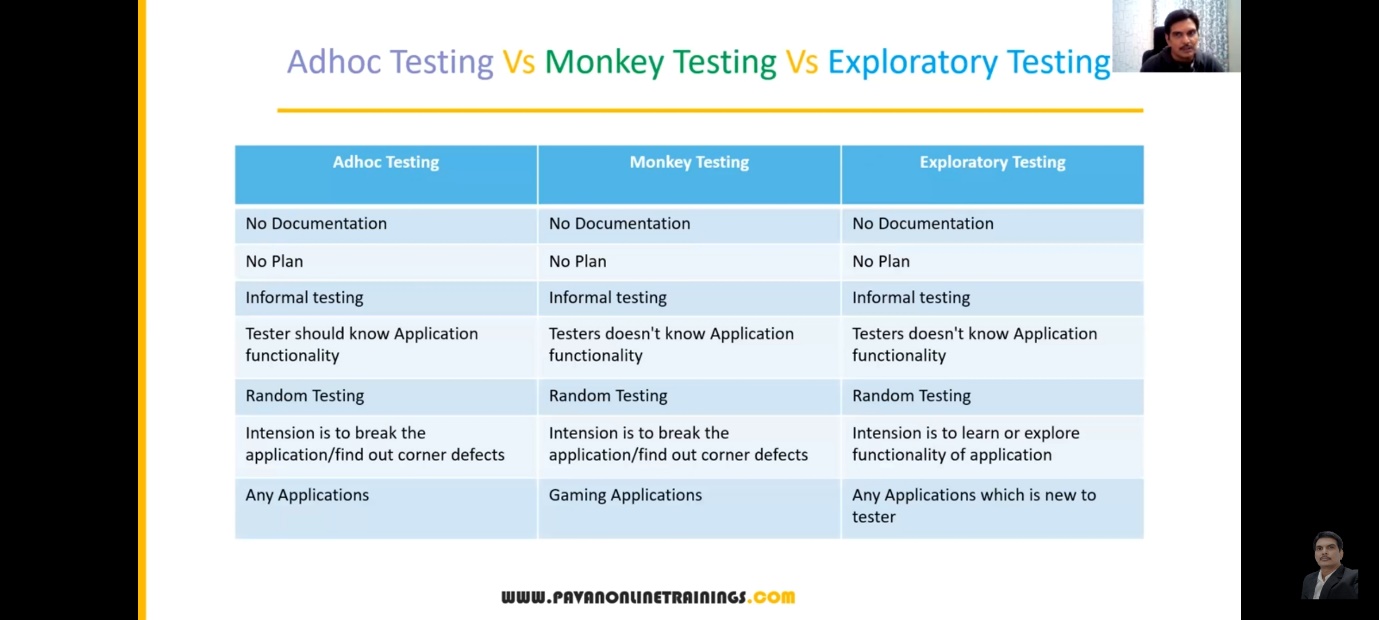
Adhoc Testing :-



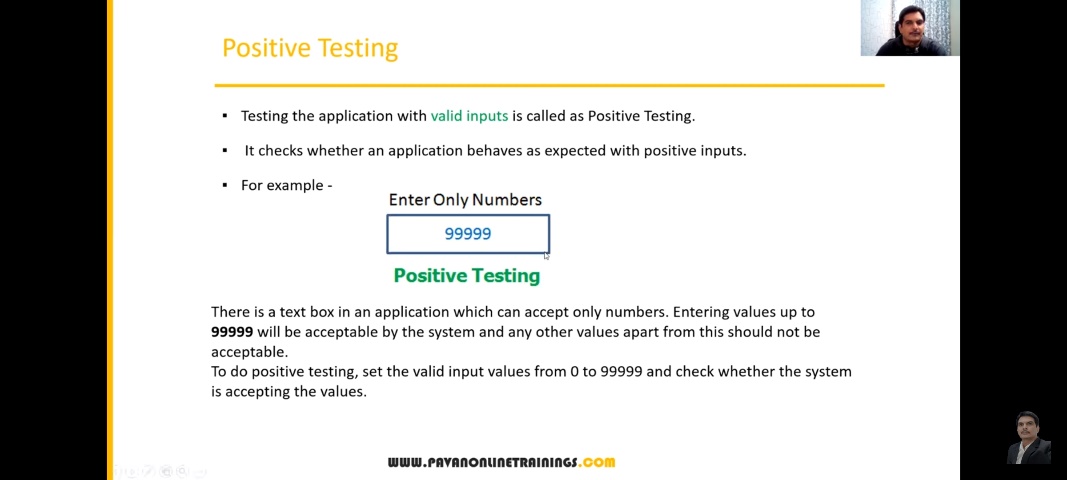
Monkey/Gorilla Testing :-



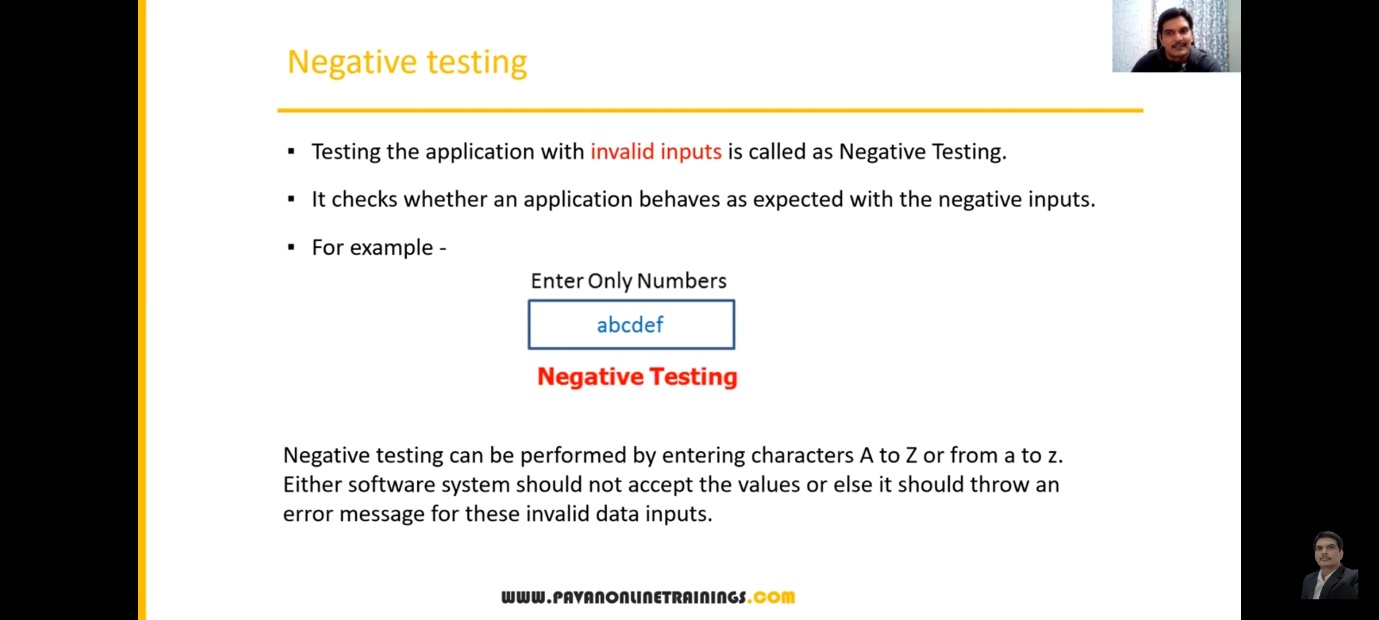
Adhoc Testing Vs Monkey Testing Vs Exploratory Testing :-



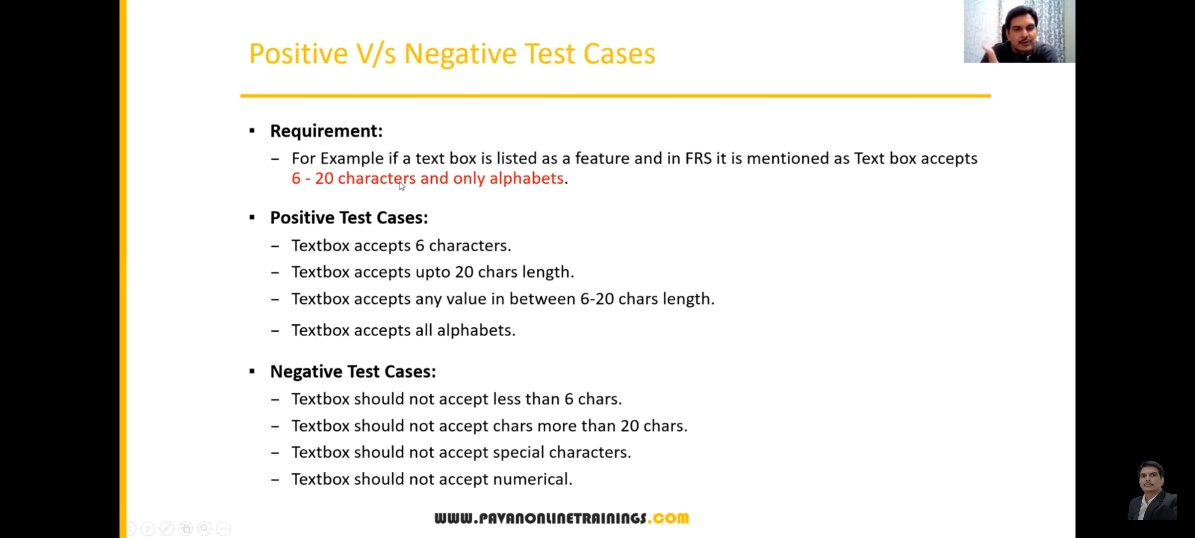
Positive Testing :-



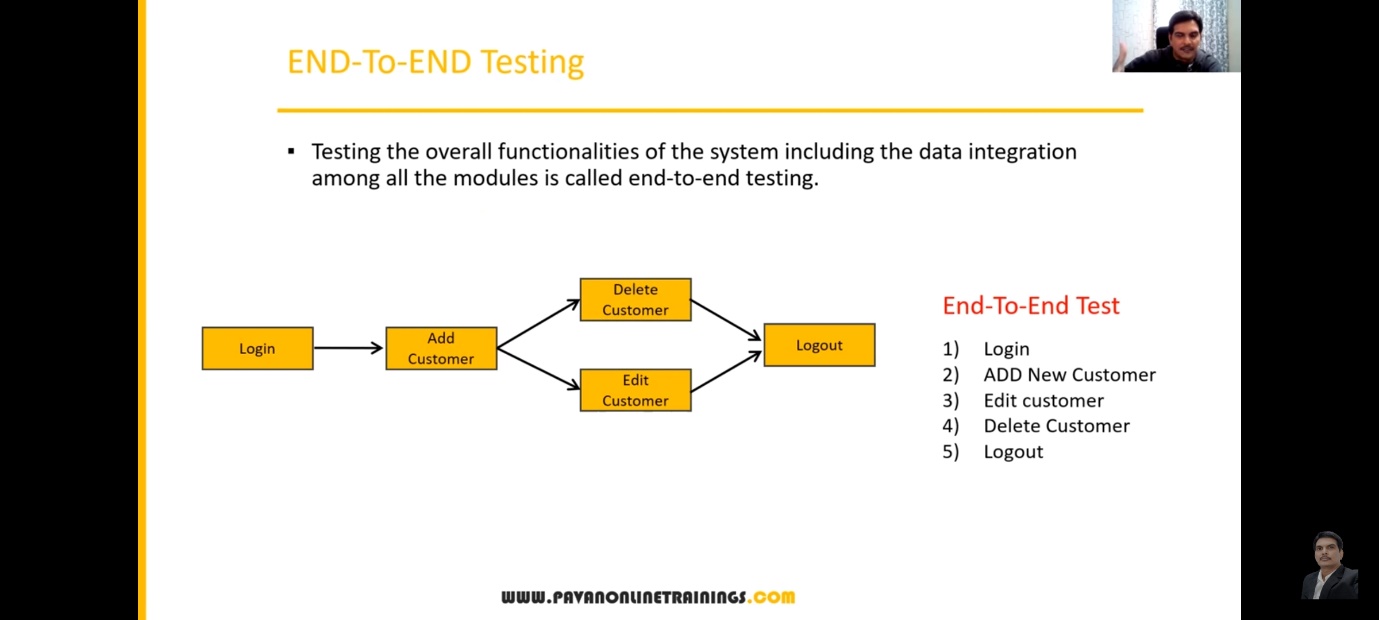
Negative Testing



Positive Vs Negative Test Cases :-



End-To-End Testing –



Globalization and Localization Testing :-



Session – 6

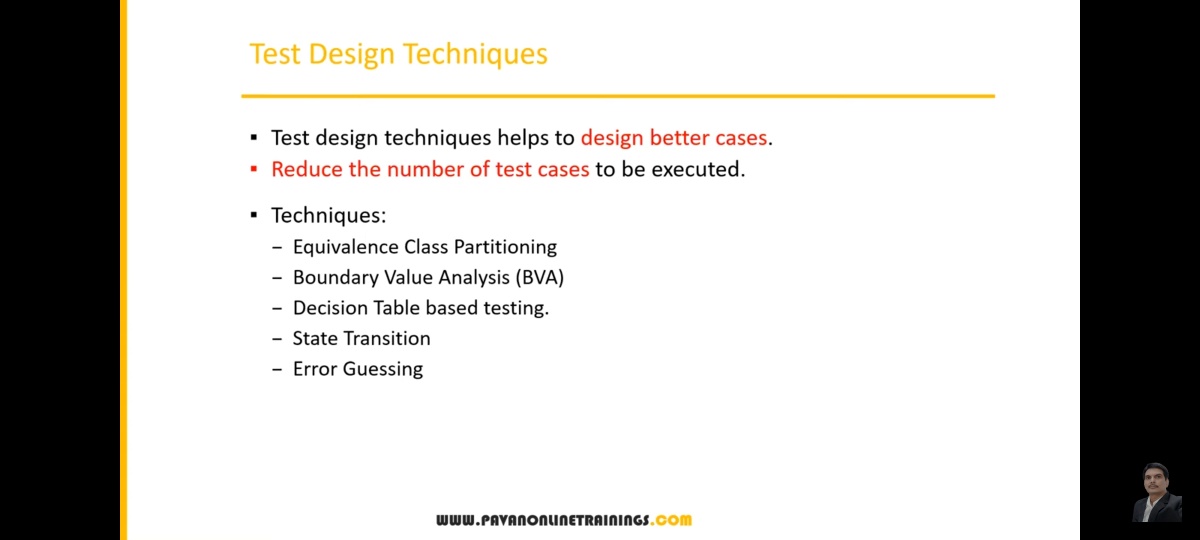
Test Design Technique /Test data design technique/test case design techniques:- It is used to prepare data for testing.

Advantage –

1. Reduce the data
2. More Coverage

Type :-

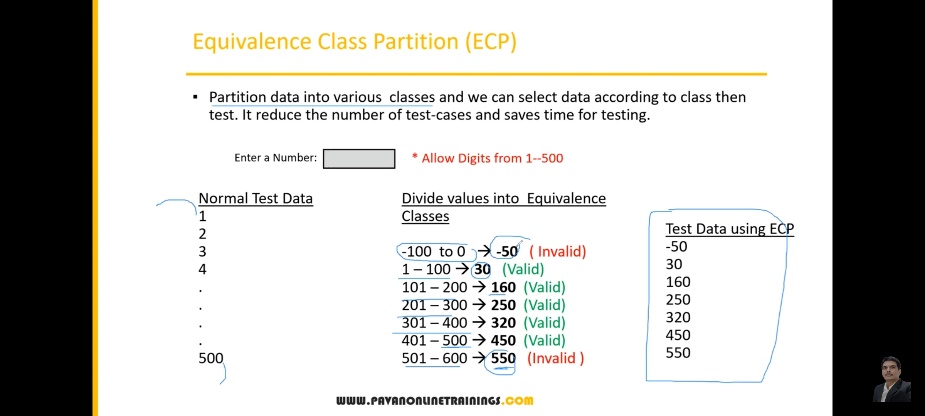
1. Equivalence class partitioning
2. Boundary value analysis (BVA)
3. Decision Table based testing.
4. State Transition
5. Error Guessing

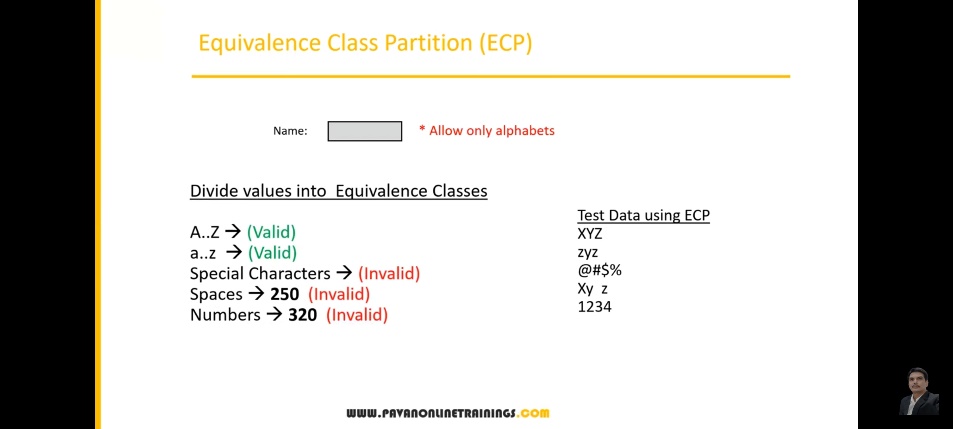


**Equivalence class partitioning (ECP):-**

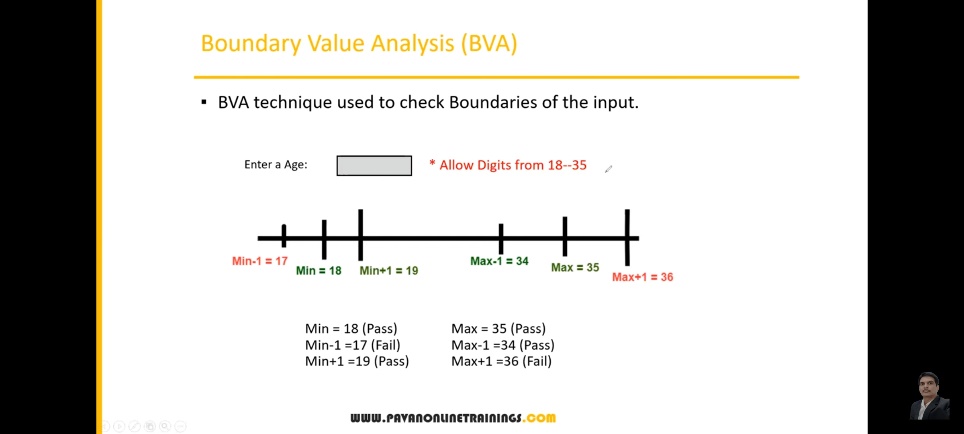
check value

classify / devide / partition the data in to multiple classes.





**Boundary value analysis (BVA) :-** BAV technique used to check Boundaries of the input.



min max

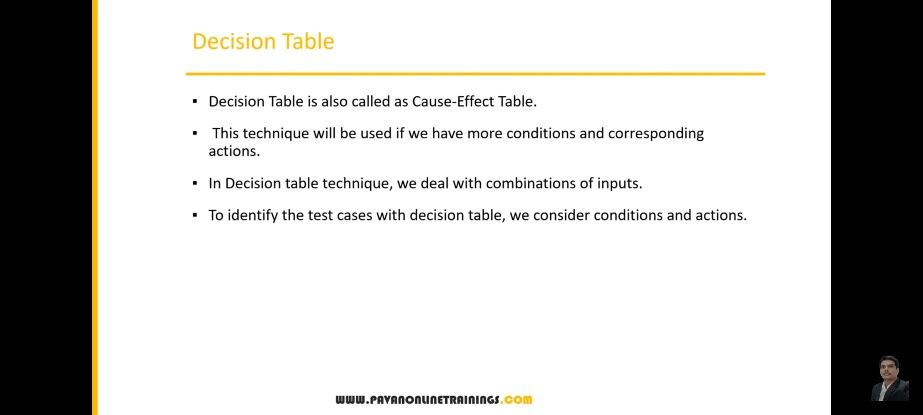
min+1 max+1

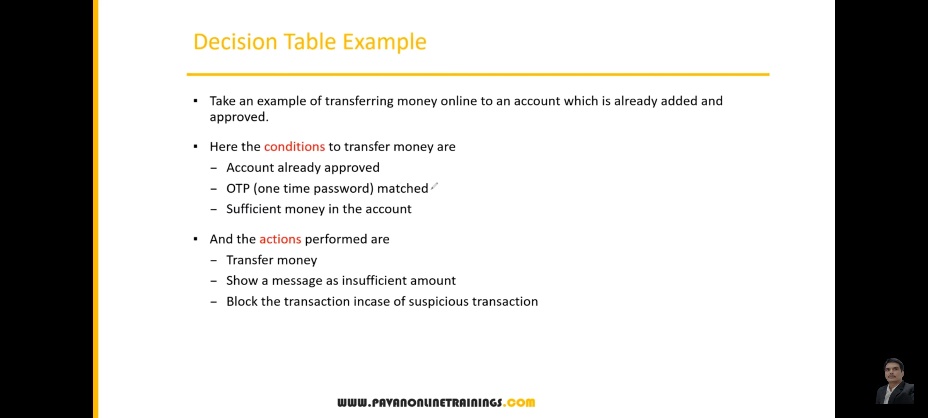
min-1 max-1

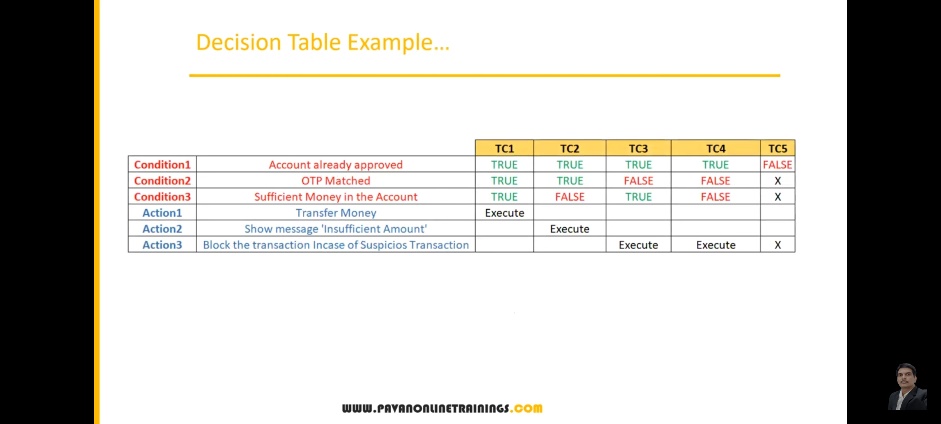
NOTE = ECP and BVA basically used in **Input domain testing**

**Input domain technique** :- The value will be verified in the text box/input fields

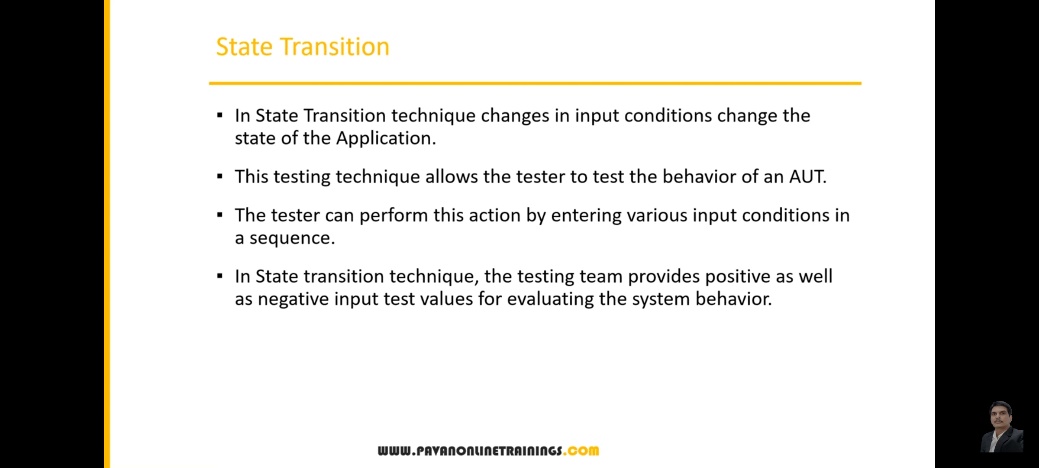
**Decision Table :-**  if we have more number of condition / actions then we use decision table technique.





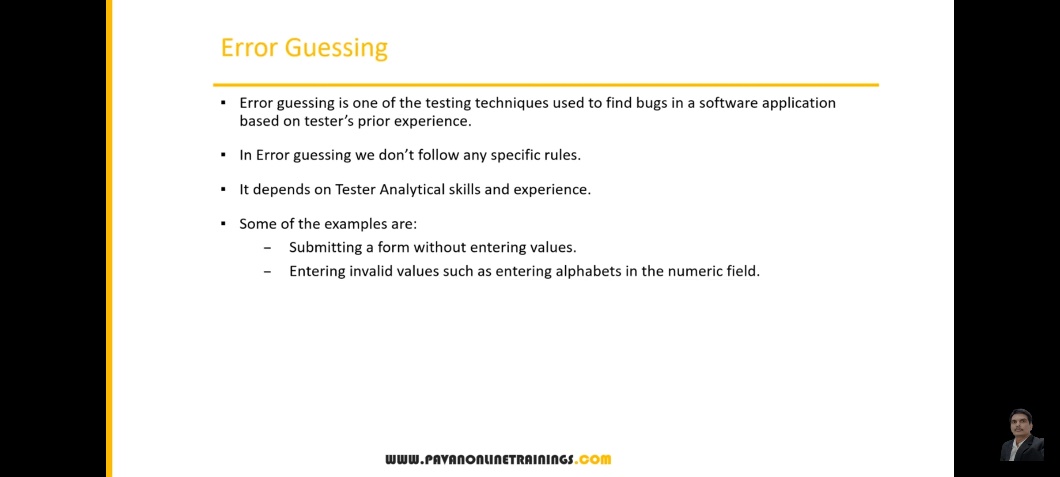


**State Transition**

****

****

**Error Guessing**



Session – 7

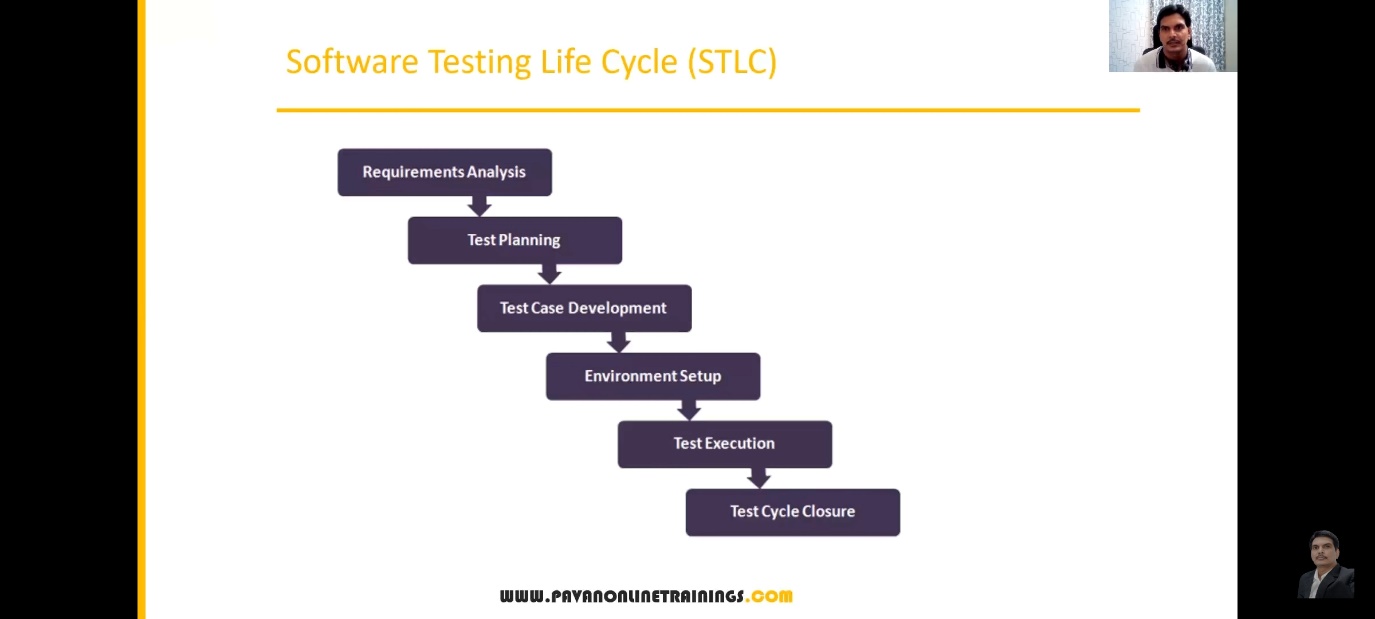
SDLC – Software Development Life Cycle

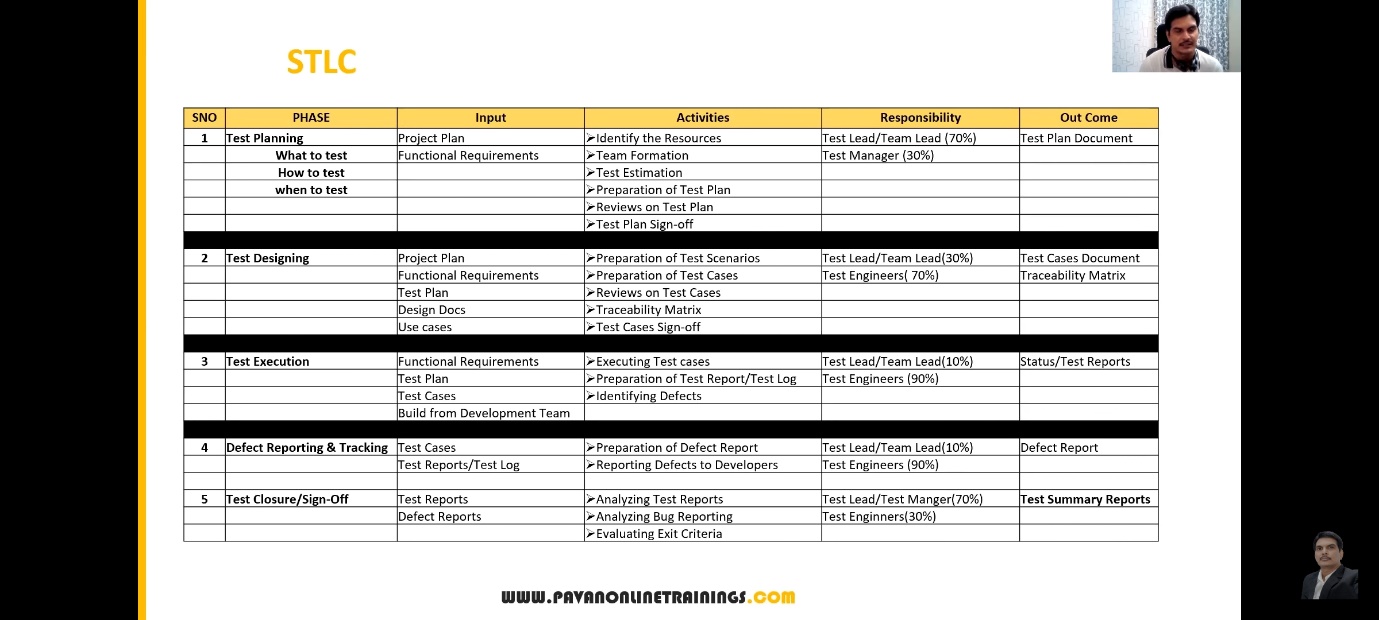
Requirement , design, coding ,testing, deployment, maintenance

STLC – Software Testing Life Cycle

1. Requirement Analysis (Understand the functionality)
2. Test planning
3. Test Design
4. Test Execution
5. Defect Reporting / Bug Reporting & Tracking
6. Test Closure

Test Scenario = it means what to test





Session – 8

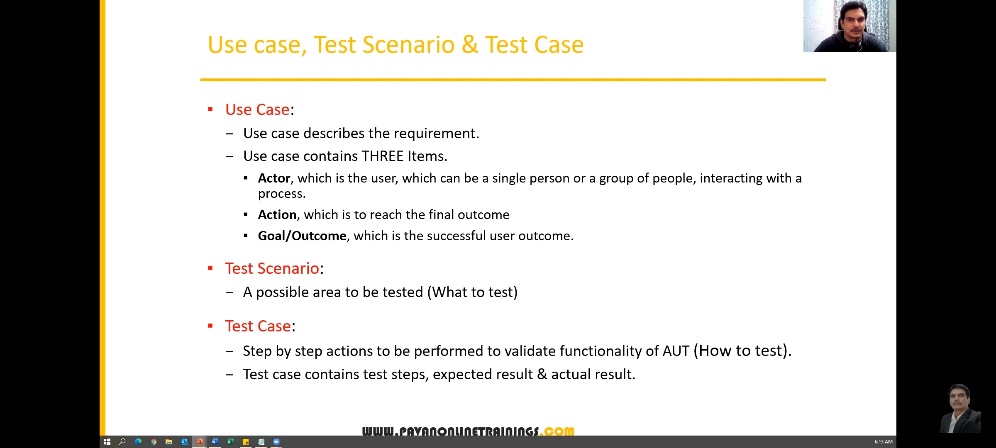
Test Plan Contents

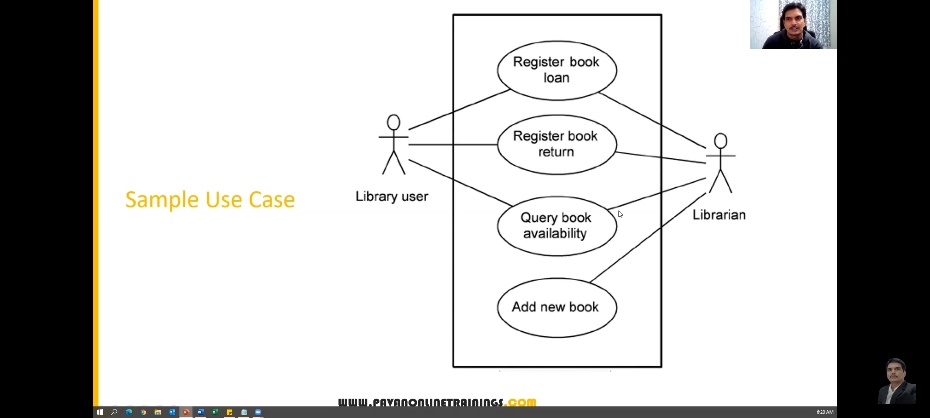
A test plan is a document that describes the test scope, test strategy, objectives, schedule, deliverables and resources required to perform testing for a software product.

Test Plan template contents :

* Overview
* Scope
  + Inclusions
  + Test environments
  + exclusions
* Test Strategy
* Defect Reporting Procedure
* Roles / Responsibilities
* Test Schedule
* Test Deliverables
* Pricing
* Entry and Exit Criteria
* Suspension and Resumption criteria
* tools
* risks and mitigations
* Approvals

Use case, Test Scenario & Test case :





Use Case Vs Test Case :

* **Use Case –** Describes functional requirement, prepared y business analyst(BA)
* **Test Case –** Describes Test Steps/Procedure, prepared by Test Engineer.

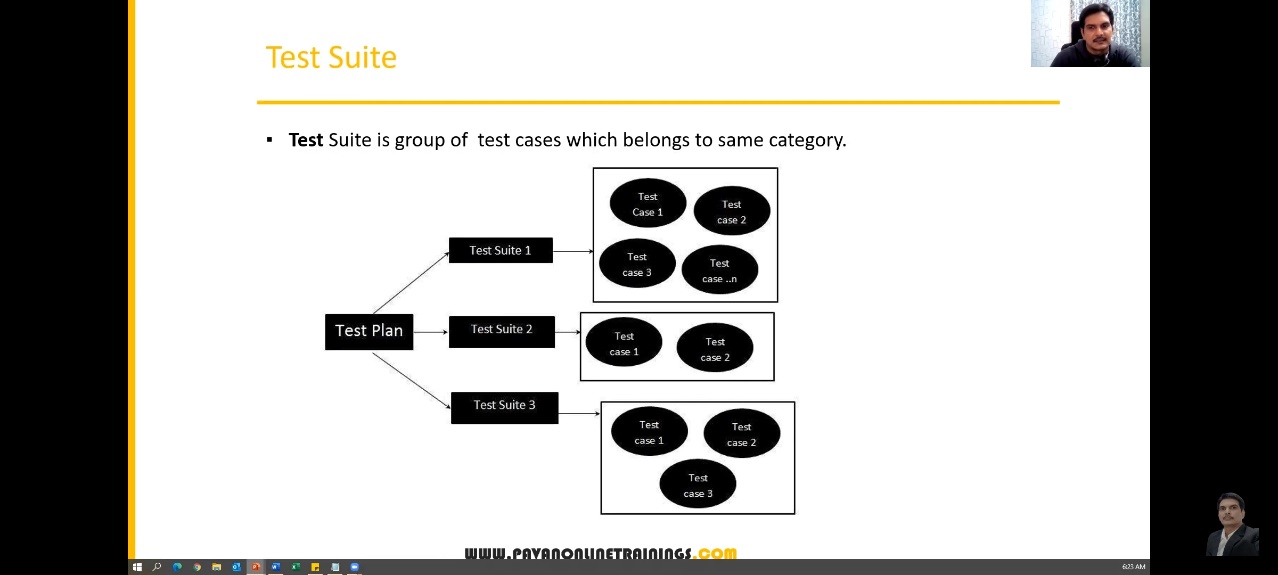
Test Scenario V/S Test Case :

* Test Scenario is ‘what to be tested’ and test case is ‘how to be tested’

Example –

* Test Scenario : Checking the functionality of Login Button
  + - TC1: click the button without entering user name and password.
    - T C2: click the button only entering user name.
    - TC3: click the button while entering wrong user name and wrong password.

Test Suite :-



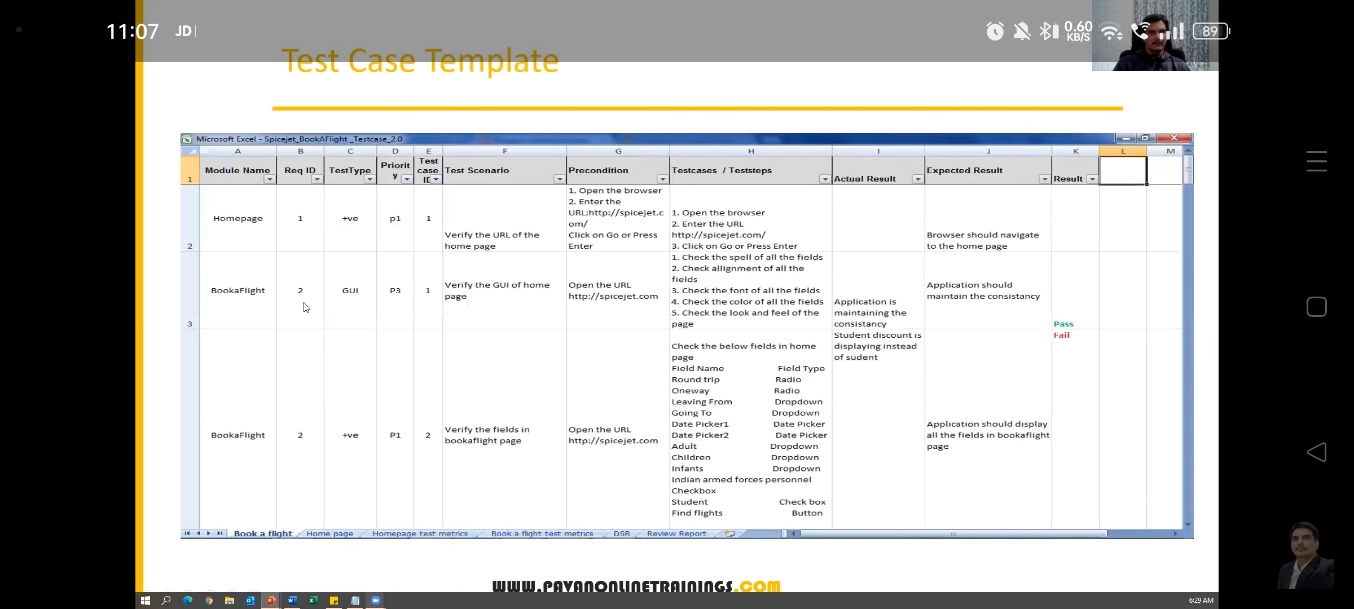
What is Test Case :-

* A Test Case is a set of actions executed to validate particular feature or functionality of your software application.

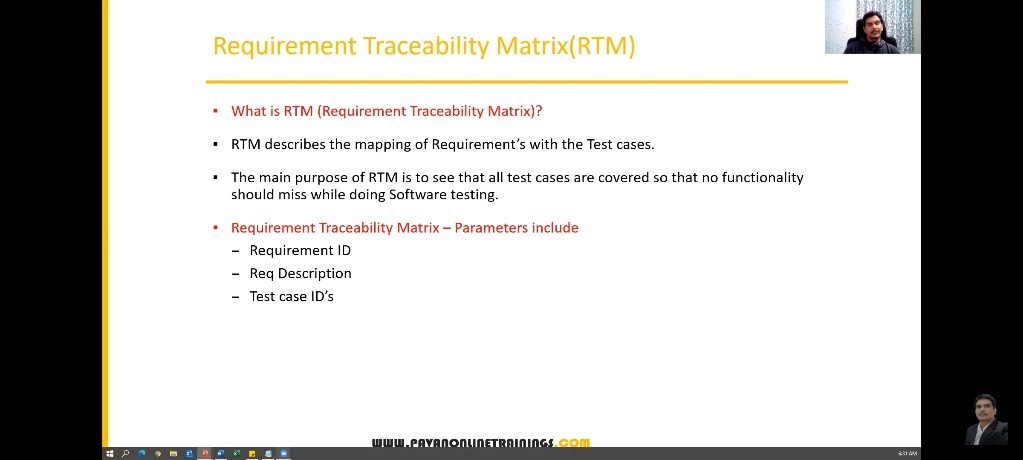
Test Case Contents –

* Test Case ID
* Test case title
* Description
* pre-condition
* priority (P0, P1, P2, P3) – order
* requirement ID
* steps / Actions
* Expected Result
* actual result
* test data

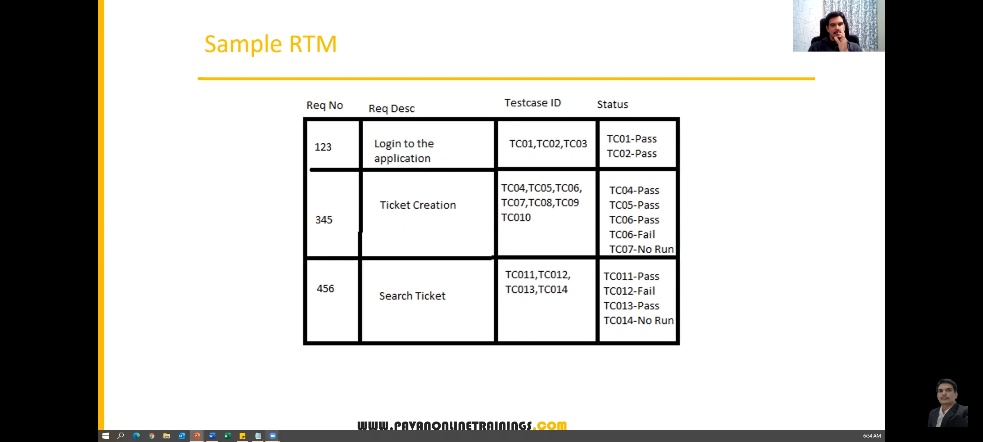
Test Case Template :



Requirement Traceability Matrix (RTM)



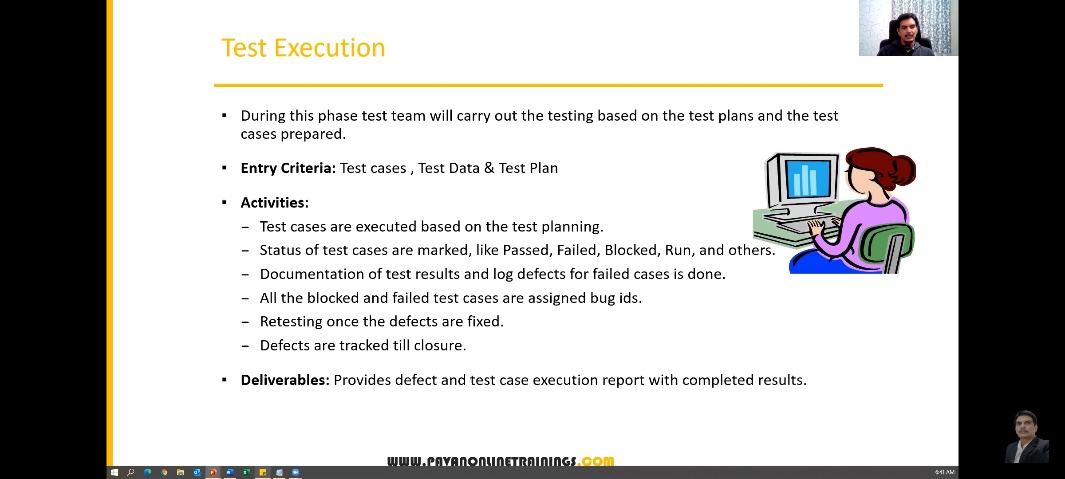
Sample RTM –



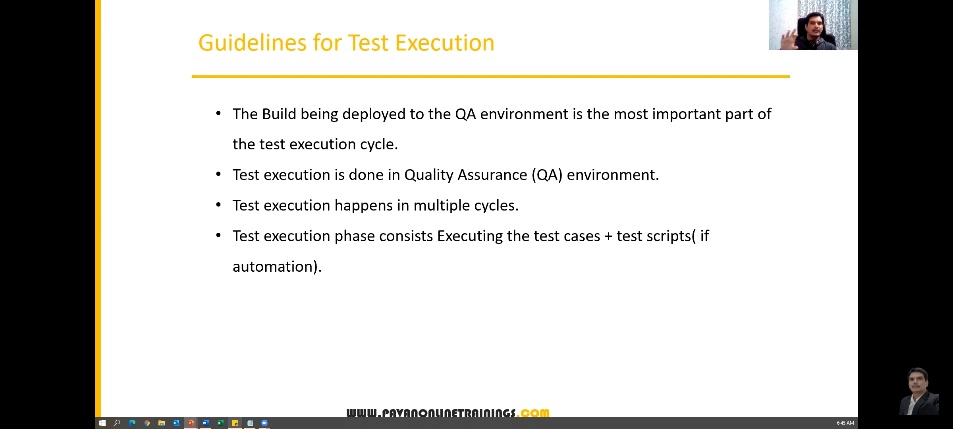
Test Environment –

* Test Environment is a platform specially build for test case execution on the software product.
* It is created by integrating the required software and hardware along with proper network configuration.
* Test Environment simulates production/real time environment.
* Another name of test environment is Test Bed.

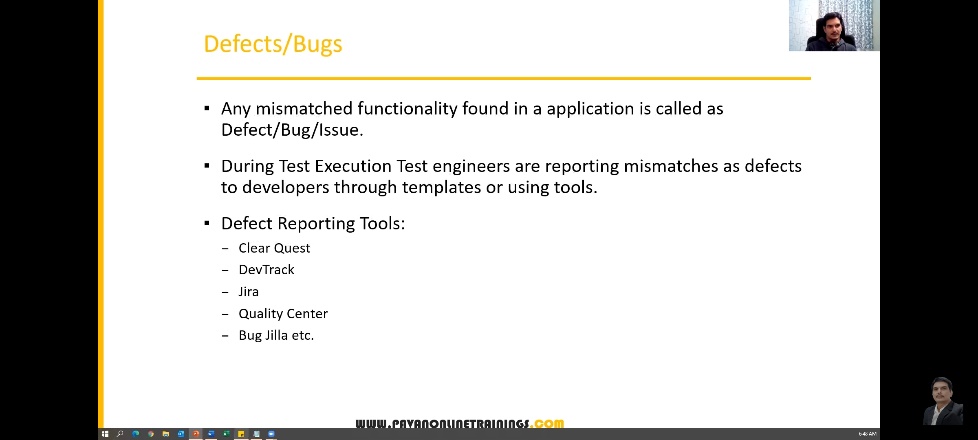
Test Execution –



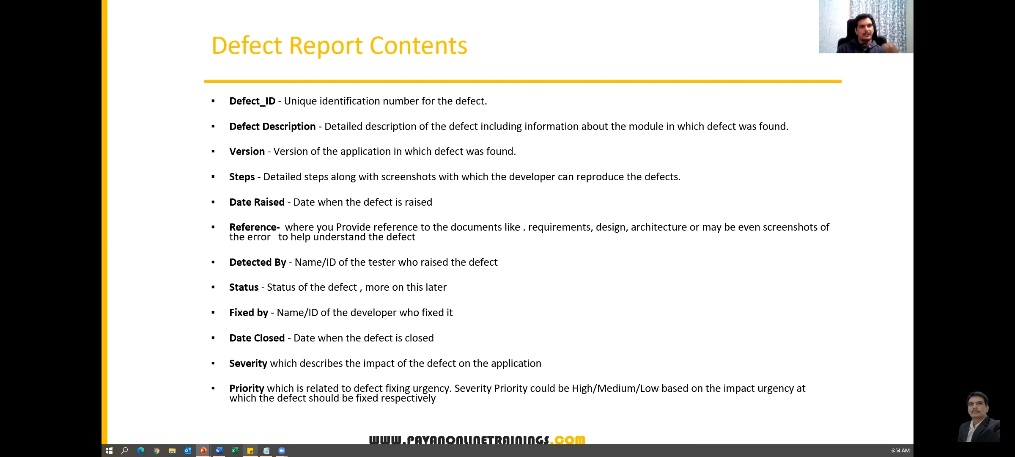
Guidelines for Test Execution –



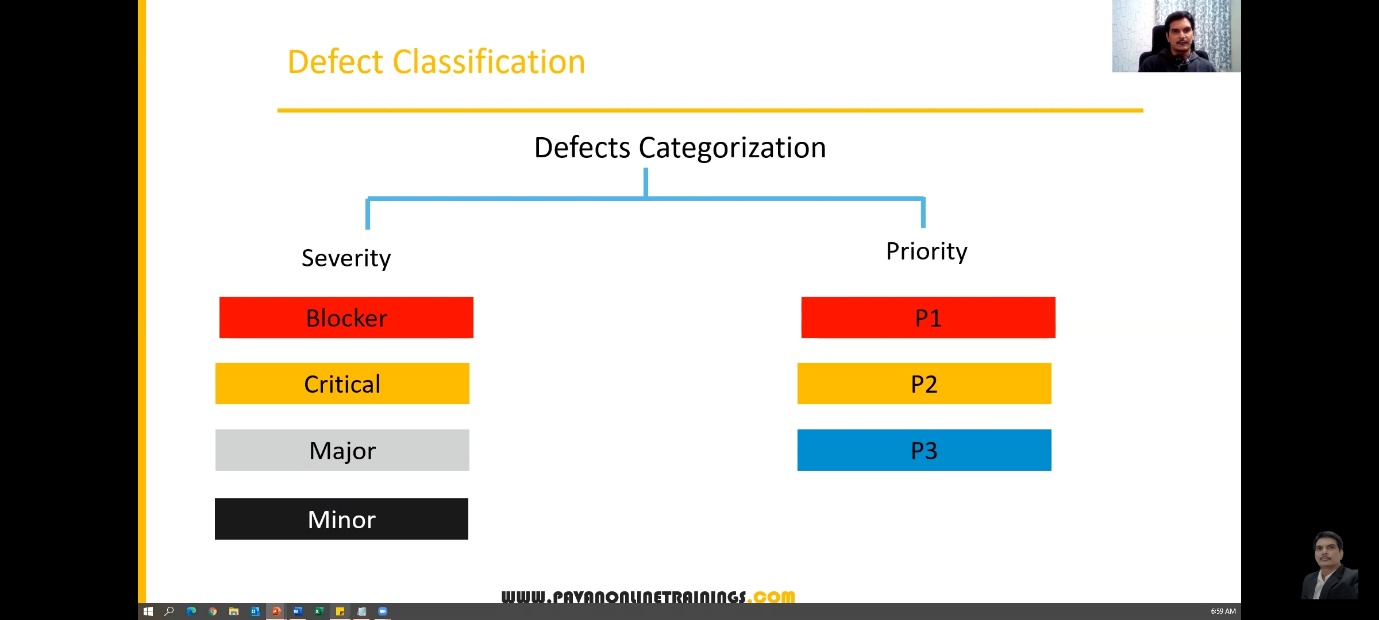
Defects / Bugs :



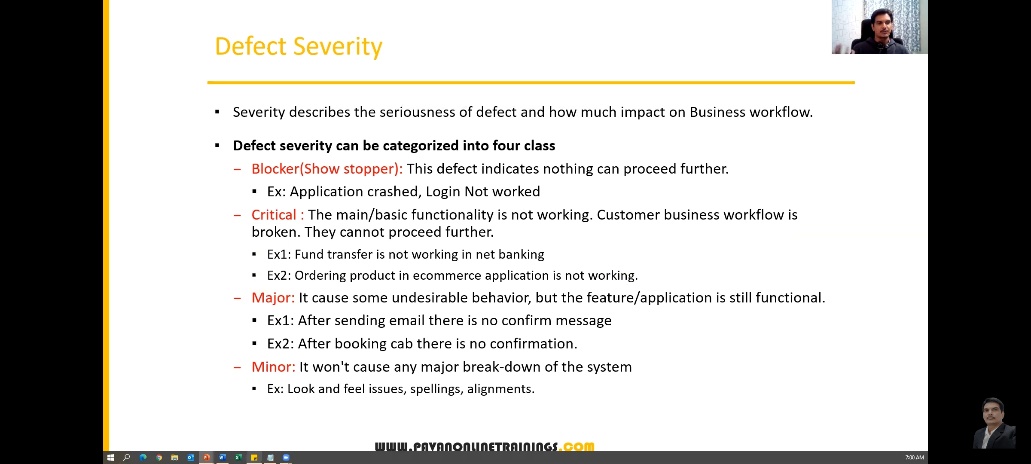
Defect Report Contents :



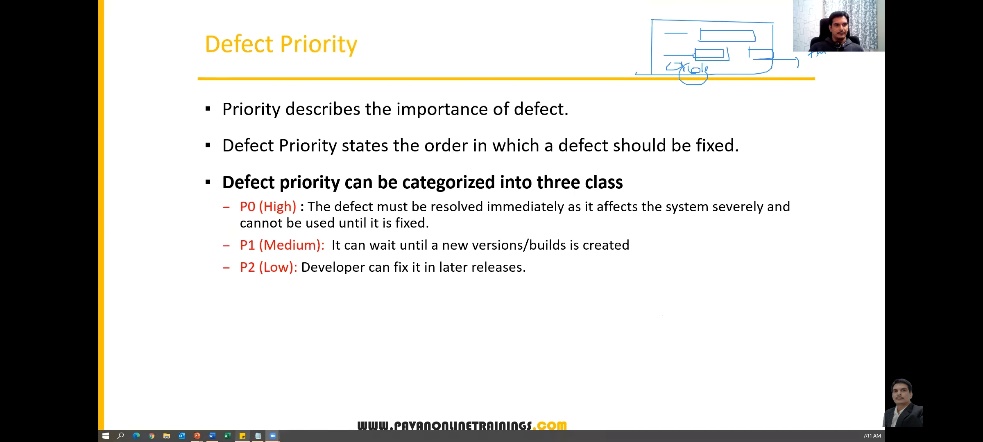
Defect Classification –



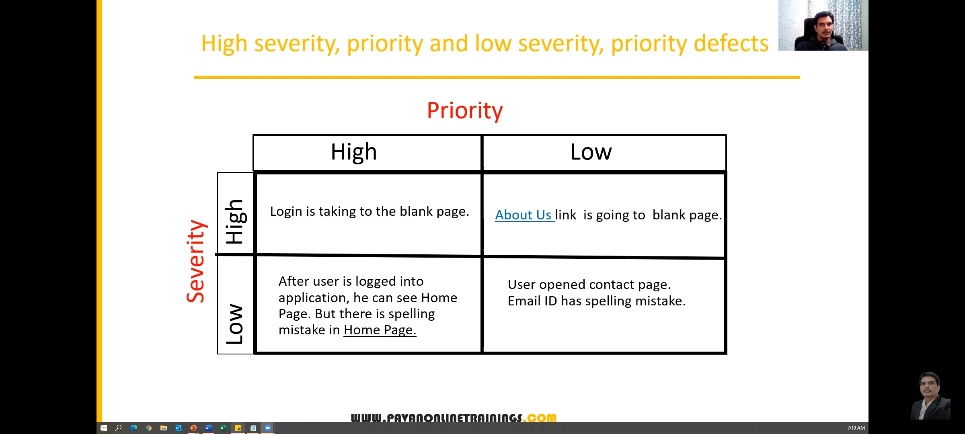
Defect Severity :-

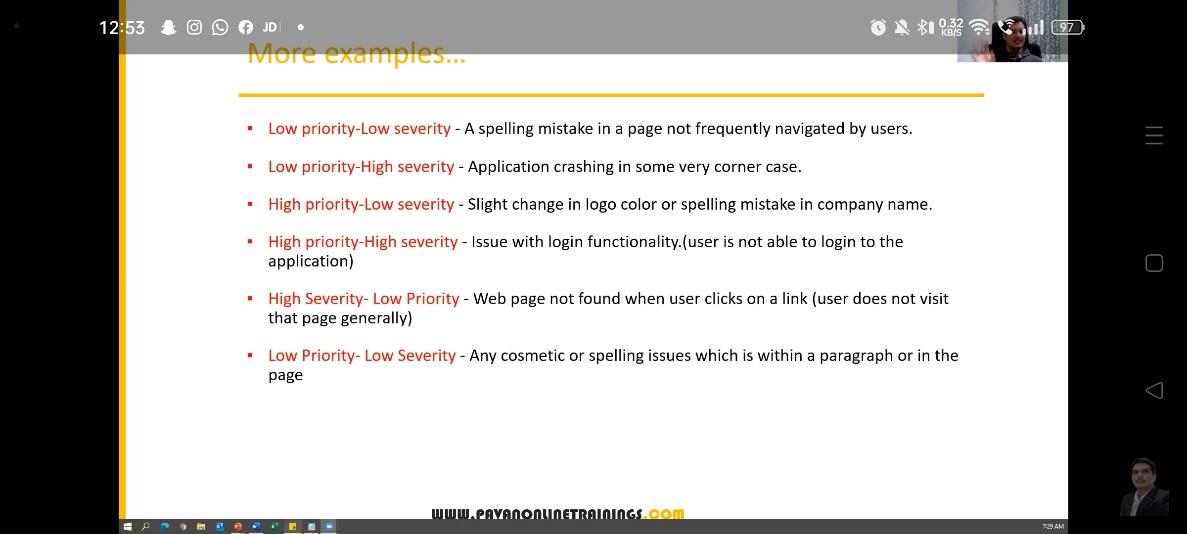


Defect Priority –

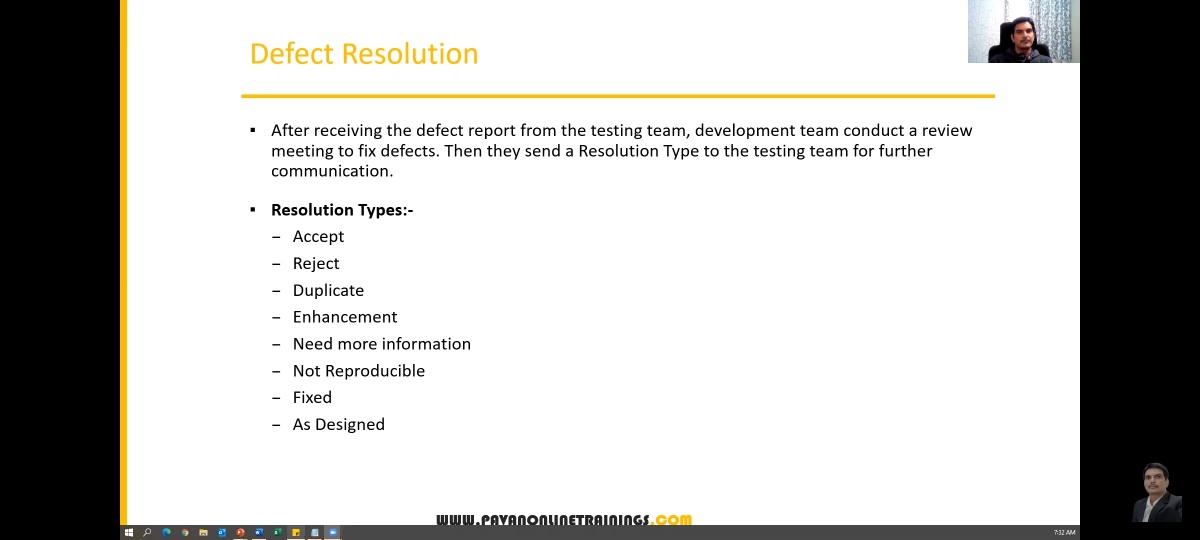


High severity, priority and low severity, priority defects –





Defect Resolution –



Session – 9

Defect/Bug Life Bycle –

defect life cycle is always says the state of the defect

