[https://github.com/DEEPAKKAMDE/BalajiCompleteTrainingData](https://github.com/DEEPAKKAMDE/BalajiCompleteTrainingData" \t "_blank)

https://drive.google.com/drive/folders/1e2l4PZuFNlIqMAejXj8e56-Wd5V5t2Xo

Software Testing (Manual + Automation)

Software Testing (Manual Testing)

Software: -

* A software is a collection of Computer Program that helps us to perform a task.

Types of Software: -

1.System Software Example: - Device drivers, Operating System etc.

2.Programming Software Example: - Compiler, Interpreter, Debuggers etc.

3.Application Software. Example: - Web Application, Mobile Application, Desktop Application.

ABC BANK(Company) ----> IT Company ---> Develop > Testing> Deliver ---> ABC Bank ----------

Software Quality: -

* Bug-Free
* Delivered on time.
* Within Budget.
* Meet requirements and/or expectations.
* Maintainable

Project vs Product: -

* If software application is developed for specific customer based on the requirement, then it is called as Project.
* If Software application is developed for multiple customers based on market requirement, then it is called as Product. Service Based Company

: - IBM, Infosys, Wipro. Product Based Company: - Google, Adobe, Oracle

Defect/Bug, Error, Failure:

1.Defect: -Deviation from the requirement specification is called as defect.

If the feature is not working according to the requirement specification is called as defect.

2. BUG: - Informal name given to defect is called as bug.

3. Error: -Mistake done in the code which is not allowing you to compile or execute is called as error. A. Compile Time Error B. Run Time Error

4. Failure: - A defect or bug will lead you to failure. #. A defect found in the production environment/ client environment is called failure.

Q. Why the Software has bugs?

* Miscommunication or No Communication.
* Software Complexity.
* Programming Error.
* Changing Requirements
* Lack of skilled tester.

Browsers: -

1. Google Chrome

2. Firefox

3. Internet Explorer

4. Opera

5. Edge

6. Safari (Supported by apple)

Operating System: -

1. Linux

2. Windows

3. MacOS

Programming Language: -

1. Java 2. Python 3. JavaScript 4. PHP 5. C# 6. Ruby 7. Dart 8. Smalltalk

White Box Testing: -

#. Testing each and Every line of code is called as white box testing. #. The Testing which is done by the 'developer' is called as white box testing.

Black Box Testing: -

#. To Verify the functionality of an application against the requirement specification is called as 'Black Box Testing'. #. It is also called 'functional Testing' or 'Behavioural Testing' #. Testing which is done by the 'Test Engineer' is called as 'Black Box Testing'.

Types of Black Box Testing: -

1. Functional Testing

2. Integration Testing

3. System Testing

4. Acceptance Testing /UAT Testing

5. Performance Testing

6. Smoke Testing

7. Ad hoc Testing

8. Exploratory Testing

9. Compatibility Testing

10.Usability Testing

11. Accessibility Testing

12. Reliability Testing

13. Regression Testing

14. Alpha Testing

15. Beta Testing

16. Localization Testing

17. Globalization Testing

18. Sanity Testing

19. System Integration Testing

As a Test engineer, we perform below testing

1. Functional Testing 2. Smoke Testing 3. Regression Testing 4. Integration Testing 5. Sanity Testing

Grey Box Testing: -

#. Combination of White Box Testing and Black Box Testing is called as Grey Box Testing.

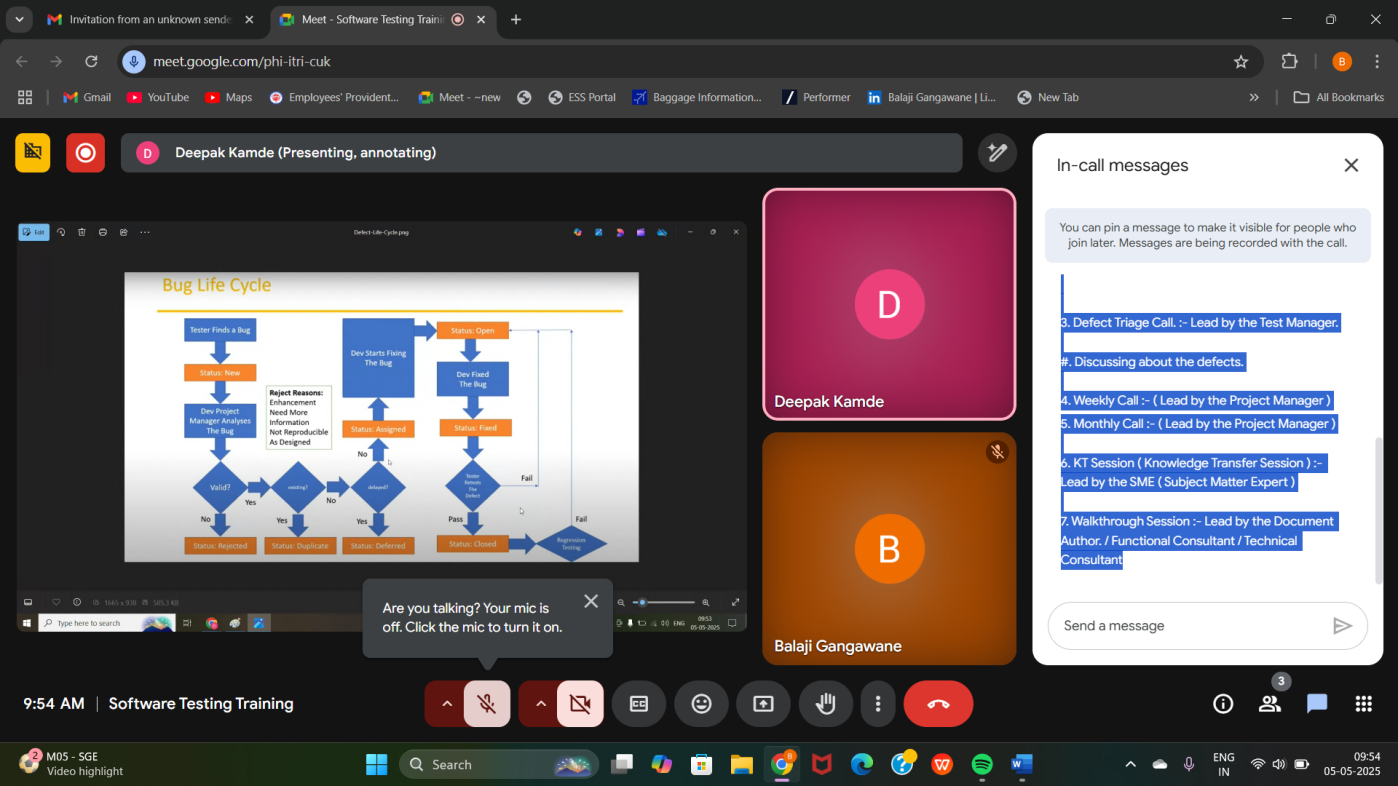
Q. Why developer shouldn't test the application?

#. Developer will never find the mistake in the code written by himself.

#. If Developer busy in the testing, then the time spend on coding will be less.

Defect Triage Meeting: - #. Test Lead #. Development Lead #. Tester #. Developer #. Project Manager (Optional)





Q. Tell me about your daily task/ daily routine in office

Example: -12 PM to 9 PM

1. Standup Call: (Daily Meeting Call): - Lead by the Test Lead.

#. What I have done Yesterday.

#. What I will do Today.

#. What is all issue I have faced yesterday.

1. Status Call: - Lead by the Test Lead

#. How much work i have completed so far.

1. Defect Triage Call: - Lead by the Test Manager.

#. Discussing about the defects.

4. Weekly Call: - (Lead by the Project Manager)

5. Monthly Call: - (Lead by the Project Manager)

6. KT Session (Knowledge Transfer Session): - Lead by the SME (Subject Matter Expert)

7. Walkthrough Session: - Lead by the Document Author. / Functional Consultant / Technical Consultant

Q. Why do they reject the defect.

#. By Misunderstanding. (In between Tester and Developer) #. #. Log In / Login.

Q. Why do we get Postpone Status?

#. Whenever the Test Engineer (TE) send a minor bug and the developer will busy in fixing the critical bug. Then we get a status a postpone.

Q. Duplicate Status: - #. Test Engineer (TE) will find the bug and send it to developer. Developer will mark this defect as Duplicate.

Q. Why do we get duplicate status.

#. Because someone else already sent the same bug.

Non-Producible Bug: -

#. The Bug which is seen by the test Engineer but not by the developer is called as non-reproducible bug.

RFE: - (Request for Enhancement)

#. The feature that are not present in the application, that could be added in enhancement that is called as RFE.

Test Data: -

#. Test Data used for testing purpose is called 'test data' Example: - username: john@gmail.com password: Xyz@1234

**Different Types of Environments: -**

**1. Dev Environment #. Development Environment used by the developers. Example: -** [**https://www.facebook.com/login/dev01**](https://www.facebook.com/login/dev01)

**2. Test Environment #. Testing Environment used by the Test Engineers. Example:-[https://www.facebook.com/login/test01](https://www.facebook.com/login/test01" \t "_blank)**

**3. UAT #. Used by the Customers Example:-[https://www.facebook.com/login/uat](https://www.facebook.com/login/uat" \t "_blank)**

**4. Production Environment #. Used by the End Users. Example:-** [**https://www.facebook.com/login/**](https://www.facebook.com/login/)

**Defect Report Parameters:**

* **Bug ID**
* **Status**
* **severity**
* **Priority**
* **Platform**
* **Module**
* **Test case id**
* **Test data**
* **Expected Result**
* **Actual Result**
* **Environment found**
* **Reporter**
* **Assigned**
* **Summary/Repo Steps**
* **Attachment/Screenshot**
* **Comments**

**Severity and Priority for defect/Bug:**

**1. Severity of bug**

* **Impact of the bug on customer's business is called severity**

**Different type of severity**

* **Blocker/Show Stopper**
* **Critical**
* **Major**
* **Minor**

**2. Priority**

* **How soon bug has to be resolve/importance of bug**

**1. Minor but High Priority Bug. #. Flipkart #. LipKart**

**2. Critical but Low Priority Bug. Username: Password: In Login Button, 'Login' Text is not Visible.**

**3. High Severity and High Priority. #. if username/password textbox is disabled. #. if Facebook application is not accessible. #. if user is not able to login.**

**4. Low Severity and Low Priority. #. Cosmetic Changes.**

**Defect Triage: - #. When the time is less and there are too many defects which are left out. then all the stakeholders (TE,TL,DE,DL,TM,DM) will meet and categorize the bug. #. They Set all the critical Bug to 'Assigned' status. #. Then all the bugs which can be fixed later to 'POSTPONE' status. #. all the left out bugs will set to cannot be fixed**

**SIT Phase 1: 1 May 2025 to 15 June 2025. 500 TC's 500/33 15 TC's Need to Execute Per Day. 5 TC's we can execute. Need 3 Resources. SIT Phase 2 UAT ORT Production.**

**Defect Masking: - #. One Bug Covering another bug is called defect masking.**

**Defect Seeding: - #. Intentionally introducing a defect to check the effectiveness of the Test Engineer is called as defect seeding.**

**Defect Leakage: - #. The Bug which is found in the 'Production' Environment or 'Customer Site' is called as defect leakage. #. When the bugs are found in the production environment, then it is sent to the Project Manager and these bugs has to be fixed within 3 hours or 3 Days. #. This**

**#. This process is called as 'Hot Fix' or 'Incident Management'.**

**Root Cause Analysis: - #. After Incident Management, meet and discuss about the cause of the defect and it is called Root Cause Analysis or 'Fish Bone technique' or ' Ishikawa Method'**

SDLC: - Software Development Life Cycle.

**#. Software Development Life Cycle is a process used by software industry to design ,develop and test software.**

**P- People**

**P- Process**

**P- Product**

**Phasea:**

* **Requirement Analysis**
* **Design**
* **Coding**
* **Testing**
* **Maintenance**

**Application Name:**

**Gmail Modules: -**

1. **Login 2. Inbox 3. Compose 4. Sent 5. Receiver Email 6. Draft 7. Setting etc....**

**Waterfall Model: -**

**Advantages: -**

**#. 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 stage.**

**#. Preferred for small projects where requirements are freeze.**

**Dis-Advantages: -**

**#. Requirement changes are not allowed. #. Testing will only start after coding.**

**#. If there is a defect in Requirement that will be continued in later phases.**

**#. Total Investment is more in terms of rework on defect is time consuming which leads to high investment.**

**Spiral Model: -**

**#. Spiral model is iterative model.**

**#. Spiral model overcome drawbacks of waterfall model.**

**#. In Every Cycle new software will be released to the customer/client.**

**#. software will be released in multiple versions. so it is also called version control model.**

**Advantages: -**

**#. Testing is done is 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.**

**Dis-advantages: -**

**#. Requirement changes are NOT allowed in between the cycle.**

**#. There is no testing in requirement and design phase.**

**V Model: -**

**#. The V-Model is a software development model which can be presumed to be the extension of waterfall model.**

**#. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase to form the typical V-shape.**

**#. The V-model demonstrates the relationship between each phase of development life cycle and the associated phase of testing.**

**The Phase of V-Model: -**

**#. The V Model consist of number of phases. The Development phase is on the left-hand side of the V.**

#. The Coding phase is at the bottom of the V.

**#. The Validation phase is on the Right-hand side of the V.**

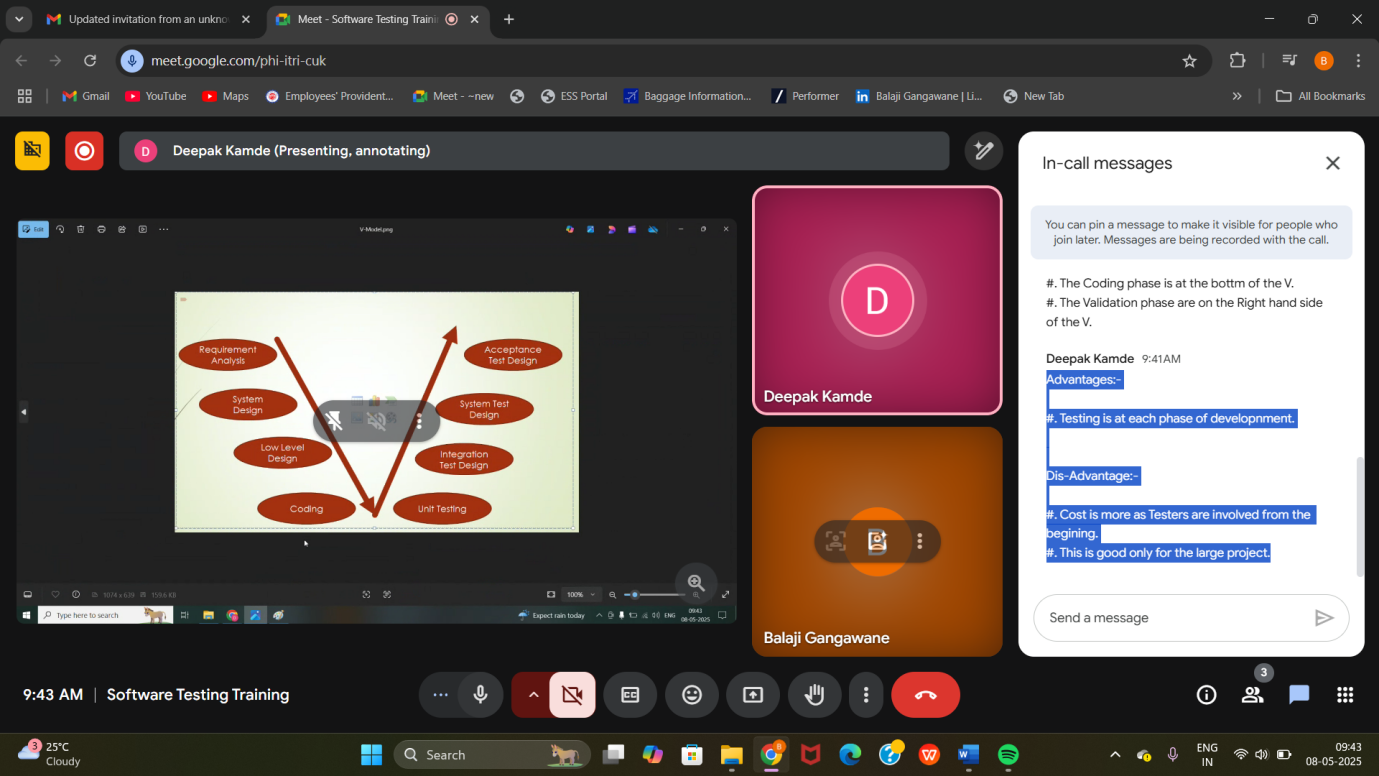
**Advantages: -**

**#. Testing is at each phase of development.**

**Dis-Advantage: -**

**#. Cost is more as Testers are involved from the beginning.**

**#. This is good only for the large project.**



**Functional Testing/ Functionality Testing: -**

* **Testing each and every "component" thoroughly is called as Functionality Testing or Component Testing or Behavioural Testing.**
* **Components: - Textbox, Text Area, Dropdown, Radio Button, Button, Label, Link/hyperlink**

**1.1 Username: -**

**1.1.1 Username text field should accept 5-32 characters.**

**1.1.2 No Special Char are allowed.**

**1.1.3 Alpha Numeric Allow**

**. 1.1.4 No Spaces should be allowed.**

**1.2 Password**

**1.2.1 Password field should accept 8-14 character**

**1.2.2 Special Char allow**

**1.2.3 Alpha numeric allow**

**1.2.4 Upper Case Mandatory**

**1.2.5 No Spaces should be allowed.**

**1.3 Mobile**

**1.3.1 It should accept 10 digits only.**

**1.3.2 It should be the Valid Mobile number.**

**1.4 Address**

**1.4.1 It should accept 12-200 Chars.**

Procedure to do functional Testing: -

* We need to test each and every component with 'Valid' data first. Then we can only go for 'invalid-data'.
* If any of the invalid data is not working then we need to check the application for the remaining set of invalid data.

Over Testing: -

Testing the same scenario in different ways is nothing but "over testing".

Example: - Username: Testing Testing123 Testing 123

Under Testing: -

Testing the application or testing the feature with insufficient set of scenarios is called 'under Testing".

Example: - Mobile: - 10-digit Mobile Number: - 9 digit

Optimized Testing: -

Testing the application with sufficient set of scenarios which make sense.

Integration Testing: -

* It is Performed when two or more functions or component of the software are integrated to form a system.
* It is basically checking the proper functioning of the software when components are merged to work as a single unit.
* Testing the Data flow or interface between two modules is called integration testing

System Testing: -

* It is an End-to-End Testing where testing Environment is similar to the Production Environment.
* End-to-End Testing Navigate through the feature and check the end feature is working or not.

Q. When do we go for System Testing?

* When we have Testing Environment similar to the Production Environment.

Q. When we do System Testing?

* When all the End-to-End features are ready.
* When all the basic feature are functionally stable.

Acceptance Testing: -

* It is an End-to-End Testing done by the Engineers sitting at customer place is called a Acceptance Testing.

Q. Why we do Acceptance Testing?

* Under business pressure the developer is push the code with lots of bugs.
* Bug in the software will cause loss of business thus we need to do acceptance testing.
* Developer might misunderstand the requirement and developed the wrong feature. To check that the customer/client have to do Acceptance Testing.

Regression Testing: - (VVVIP)

* Regression Testing is a type of testing that is done to verify that a code change in the software doesn't impact the existing functionality of the product(software).
* This is to make sure that the Product (software) works fine with the new functionality, bug fixes, or any changes in the existing feature.
* We execute test cases over and over to verify that

Data Refresh Activity (till deployment): -

Mock Data 1: 100 Accounts (Test Data) (Provided by Conversion Team): 01-Jan-2025

Mock Data 2: Next 300 Accounts (Test Data) (Provided by Conversion Team) 01-March-2025

* Will Perform Regression Testing before start actual Testing.

Q. When we do Regression Testing?

* When new functionality is added to the application.
* When there is a defect fixed.
* When there is a data refresh activity performed.
* When patch is applied.

Re-Testing: -

* After bug get resolved we retest.

Development Environment Testing Environment UT Enviroment

DE and DL TE and TL Client

FT FT, IT FT, IT FT, IT, ST

FT: Functional Testing

IT: Integration Testing

ST: System Testing

* Whenever DE (Development Engineer) developed a module then DL (Development Lead) will create a 'Build' (B01) and inform to the TL (Test Lead), TL will make sure that Build should be present in Testing Environment.
* TE (Test Engineer) will find a bug and will raise a defect. At the same time Development Engineer will developing a second module.
* After that DL (Development Lead) will create a Build (B02) and inform the TL. TL will make sure that the Latest Build should be present in the testing Environment.
* After all Module are Ready DL will inform to the TL.
* TL will contact to the Customer for providing the Server for the Testing Purpose.
* After that TL will make sure that the Latest Build from Development Environment will be Present in the UAT Environment.
* Where if it is OK then the Product will be Release by the PM (Project Manager) / Delivery Manager (DM) / Build Manager/Release Manager.
* After that client do the Acceptance Testing, if all is OK the will 'launch' the application and is called as "1st Release".
* If the application contains Minimum Bug (30-40) then it is a beautiful application.
* For launching an application, it takes around 40-90 test cycles.

Release: -

#. Starting from Requirement Gathering, developing the application, testing it for many cycles and launching it into production is called as "1st Release".

Build: -

#. The Program's (Code) written are compiled and create it's binary and these binaries are compressed and the compressed files is called as "Build".

Patch: -

#. Patch is the small software that contains modified programs. Modified Programs means addition of module, deletion of module or medication of module.

Q. As soon as you get a new Build which module will you test first?

1. Will Test the New Module.

2. Then will test the Integration between old and New Module.

3. Then We Retest the bugs of Old Module.

4. Will Retest the Entire Old Module.

Q. Why do we find bugs in OLD Module?

#. Adding one module might introduce a bug in another module.

#. Fixing one bug might introduce bug in another module.

#. Chances are there that we might have miss the bugs in previous cycle.

Test Cycle: -

#. The Duration to start the testing of the application to Completely Tested is called as one Test Cycle.

#. Test Cycle duration depends on size of the application, complexity of the application and the number of Test Engineer's are Present. 01-JAN-2025 to 23-JAN-2025 500 TC's 5 Test Engineer 6/ Day: for Each Tester

Alpha Testing: -

#. Testing done before releasing the application to the customer is called Alpha Testing. #. It is done by the Test Engineers.

Beta Testing: -

#. Testing done After releasing the application is called as Beta Testing. #. It is done by the End User.

Comparison Testing: -

#. Testing done between two competing applications where we compare two competing applications.

#. We get few advantages and few disadvantages, all the advantages send to the "marketing team", and all the drawbacks (RFE) are sent to the development team.

Smoke Testing / Build Verification Testing/ Dry Run

#. Testing the basic and critical feature of an application before we do "through" testing is called as smoke Testing.

Application WhatsApp Feature To Do Smoke Testing

#. Chat/Message

#. Setting

#. Status #. calls-Video/Audio

#. Privacy.

Smoke Testing Not Required

#. Invite a Friend.

#. File Sharing #. Profile Pic

Reliability Testing: -

#. Testing the functionality of an application for a particular period of time it is called as 'Reliability Testing'.

System Integration Testing

#. Testing the Data Flow between two "software System" is called System Integration Testing. Example: - login with Gmail/Facebook in Instagram.

Example: - Select a Payment Option ( PhonePe/Paytm/Gpay ) in Amazon.

Ad hoc Testing: -

#. Testing the application randomly without referring any formal document like Test Cases, Requirement document etc. is called as Ad hoc Testing.

Q. When we do Ad hoc Testing.

#. When the Application is functionally stable, we do ad hoc Testing.

#. At the End of Test Cycle, we do Ad hoc Testing.

Performance Testing (Non-Functional)

#. Testing the stability and Response time of an application by applying load is called as Performance Testing.

1. Load: Load is designed number of users.

2. Stability: Ability to stand with load and withhold.

3. Response Time: Time Taken to send the request , execute the Program and get the result is called as Response time. Response time: T1+T2+T3

T1 Request time

Server

Execute request T2

User

T3 Response time

Types of Performance Testing: - (Non-Functional Testing)

1. Load Testing: -

Testing the stability and response time of an application by applying load "lesser" or "equal to" the designed number of users is called as load testing.

1. Stress Testing: -

Testing the stability and response time of an application by applying load "more" or "greater" than the designed number of users is called Stress testing.

1. Volume Testing: -

Testing the stability and response time of an application by transferring 'huge volume' of data is called Volume Testing.

1. Soak Testing: -

Testing the stability and response time of an application by applying load for a 'Particular Period' of time is called as soak testing.

STLC: Software Testing Life Cycle:

Phases

* Requirement Analysis
* Test Planning
* Test Development/Test Design
* Test Environment Setup
* Test Execution
* Test Closure

Requirement Analysis (Consultant team):

* FDD -Functional Design Documents
* SRS- Software Requirement Specification
* CRS- Client Requirement Specification
* TDD- Technical Design Documents

Testing Phases:

* ST- System Testing
* SIT- System Integration Testing
* UAT
* BVT
* ORT

The main goal of STLC is to identify and document any defect or issue in the software application as early as possible in the development process.

This allow for issues to be address and resolve before the software is released to the public..

1. Requirement Analysis: -

It should be clear, consistent and testable.

1. Test Planning: -

Developing Test Strategy Document (Selecting the testing method and technique that will be used)

Identify the Test Deliverables and mile stones.

Review and Approving the Test Plan.

Assigning Roles and Responsibility to the testing team.

1. Test Cases Design/Development: -

Development of the Test Cases on the requirement.

Review & Validating of the Test Case.

1. Test Environment Setup

When the Integrated Environment is ready to Validate the product/Test Case.

1. Test Case Execution: -

We execute the test cases in the real time and try to find bugs.

1. Test Case Closure: -

Once Testing is Complete, matrix, reports are ready to share/publish with the team.

Tester Work in office

* Writing test cases/Test scenario
* Review of test cases
* Test case execution
* Raising defect
* Involve in the client meeting
* Involve in daily progress meeting
* involve in the workshop
* Involve in the walkthrough session

Test Scenario: -

* Test Scenario provides a high-level overview of what functionality to test.

Example: - 1. Verify Login Functionality. 2. Verify Logout Functionality 3. Verify Forget Password Functionality. 4. Verify User Sing Up Functionality. 5. Verify Application Access Functionality.

Test Case: -

* Test Case outline the detailed steps to execute those tests.
* Test Case is a set of Actions executed to verify a particular feature or functionality of a software application.
* A Test Case contains Test Step, Test Data, Pre-condition, Expected Result, Test Case id, Test Case Description, Action, Requirement developed for a specific test scenario.
* Test Case include a specific condition using which a test engineer can compare the expected and the actual result.

Best Practice for Writing Good Test Cases: -

* Simple.
* Keep End User in Mind.
* Avoid Repetition.
* Do Not Assume.
* Ensure 100% Coverage.
* Peer Review
* Use Testing Technique.

4-5 Test Cases Per Day. Mon-Fri: 9 Hours

2-hours: Test Data Preparation, Execution, Evidence Preparation and Upload in JIRA and pass the Test Case.

RTM (Requirement Traceability Matrix)

* It is a document through which you can make sure that every requirement has at least one test cases.

Advantages of RTM: -

* Through this document we can make sure that every requirement has at least one test case written.
* If the customer changes the requirement, then we know which test case and which automation script need to change.

Types of RTM: -

1. Forward Traceability Matrix: -

* Mapping from the 'base document' to derived document i.e. Requirement (FDD) to 'Test Case' to Automation Script is called Forward Traceability Matrix.

1. Backward Traceability Matrix: -

* Mapping from 'derived document' to 'base document' i.e. Automation Script to Test Cases to Requirement is called as Backward Traceability Matrix

Requirement>Test Cases > Automation Script: Forward Traceability Matrix Automation

Script>Test Cases>Requirement: Backward Traceability Matrix

Test Execution: -

* This is the phase where we execute Test Cases and test the application.
* This is the phase where the Test Engineer will spend most of his time.
* This is the phase wherein the Test Engineer will be more productive to the organization.
* This is the phase where Test Engineer will help the developer to fixed the bugs.

Defect Tracking: -

* Execution of Test Cases means finding bugs and these bugs has to be tracked in a organized way. This is called defect tracking.

Test Execution Report: -

* At the End of Test Cycle, we prepare a test execution report.
* It contains Number of Test Cases Written, Number of Test Cases Executed, Number of Test Cases Not Executed, No of Test Cases Passed, Number of Test Cases Failed, Pass % , and Fail %.
* This is the last phase of Testing from customer point pf view.
* Test Lead will prepare a report and he will deliver to the customer/client at the last test cycle. i.e. end of the project.

Testing Phase: -

(1-Jan-2025 to 28-Feb-2025)

Number of Test Cases Written: 1000

Number of Test Cases Executed: 950

Number of Test Passed :800

Number of Test Failed :150

Test Case Pass %: (800/950) \*100 = 84%

Test Cases Failed %: (150/950) \*100 = 16%

TC Pass % Formula: (Number of Test Cases Passed/ Number of Test Cases Executed) \*100

TC Failed % Formula: (Number of Test Cases Failed/ Number of Test Cases Executed) \*100

Retrospective Meeting: -

#. At the End of Test Cycle all the Test Engineers will meet and that meeting is called as 'Retrospective Meeting'.

#. Here we mainly discuss two things that are: -

1. What are all our achievements.

2. What are all the mistakes.

#. All the achievements will be followed in the next project.

#. All the mistake should not be followed in the next project.

Achievements: - 1. Review 2. Test Case Defect Tracking 3. Test Management

Mistake: - 1. Duplicate defects were more. 2. Not Performed the Ad hoc Testing.

Scenario:

Verify Login functionality.

TC1: Verify Login functionality with Valid Credentials.

TC2: Verify Login functionality with Invalid Credentials.

Note: - 1 Test Scenario can have multiple test case

Agile Model / Agile Methodology / Agile Process

It is an Iterative and Incremental Approach.

Agile is an Iterative and Incremental Process.

Agile Principles: -

* Customer No need to wait for long time.
* We develop, test, and release a piece of software to the customer with few numbers of feature.
* We can accept/ accommodate requirement changes.
* There will be good communication between customer, Business Analyst, Developer and Tester.

Advantages: -

* Requirement Changes are allowed in any stage of development (or) we can accommodate Requirement changes in the middle of the development.
* Release will be very fast (Weekly)
* Customer No need to wait for a long time.
* It is a very good model to adapt.

Drawback: -

* Less focus on design and documentation since we deliver software very fast.

Agile Terminologies / Agile Ceremonious: -

1. SCRUM

* SCRUM MASTER
* SCRUM TEAM (PRODUCT OWNER, DEVELOPER, TESTER)

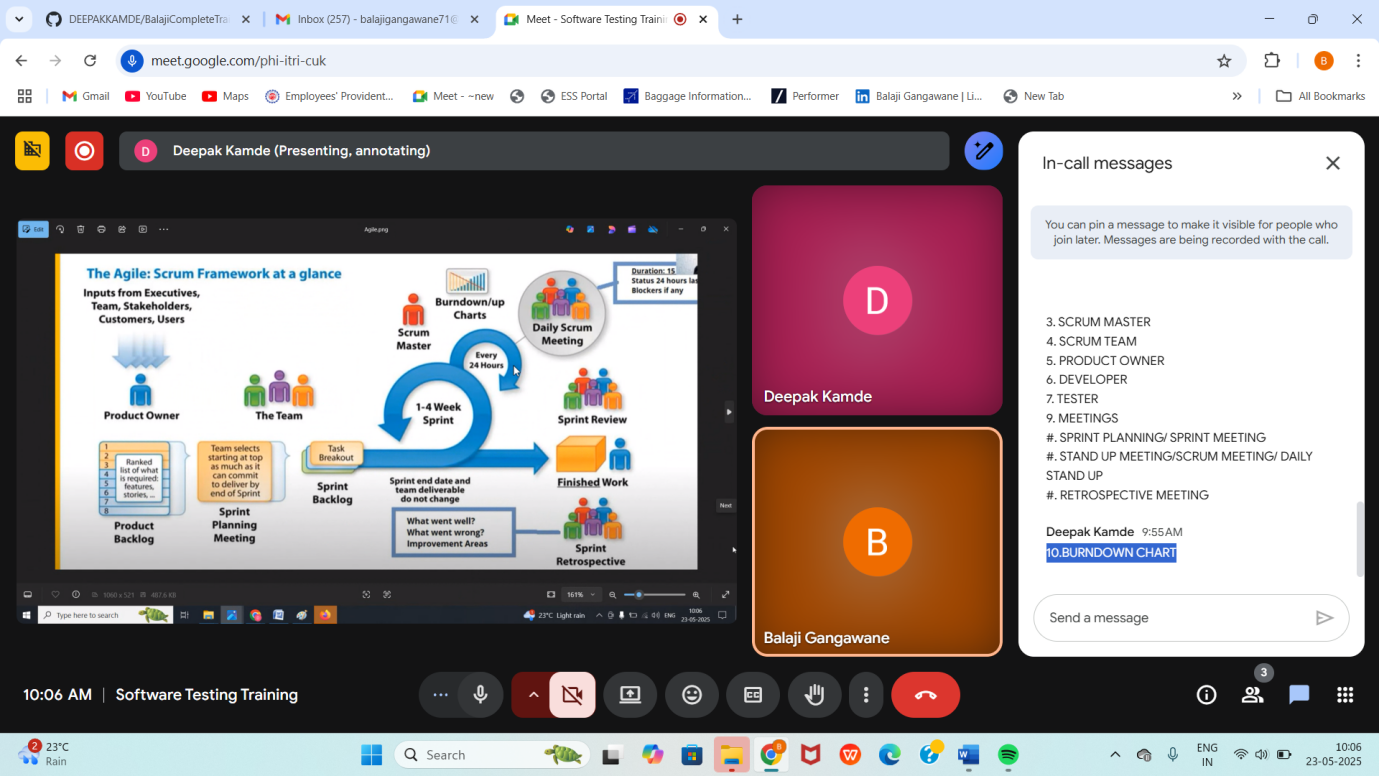
2. ARTIFACTS

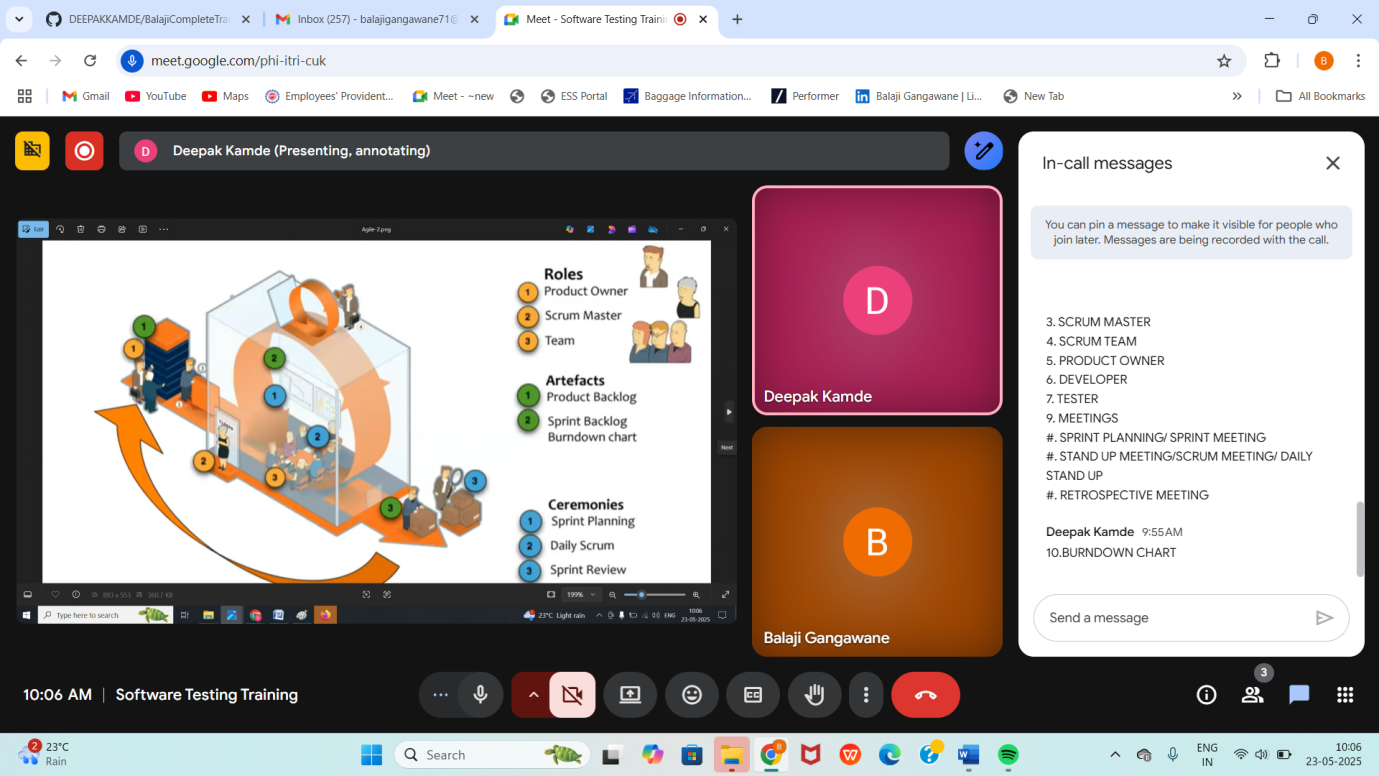
* PRODUCT BACKLOG
* SPRINT BACKLOG
* EPIC
* USER STORY/ STORY CARD / STORY
* STORY POINT
* IMPEDEMENT BACKLOG
* STORY BOARD

9. MEETINGS

* SPRINT PLANNING/ SPRINT MEETING
* STAND UP MEETING/SCRUM MEETING/ DAILY STAND UP
* RETROSPECTIVE MEETING

10.BURNDOWN CHART





SCRUM: -

* Scrum is a framework through which we build software product by following Agile Principles.
* Scrum includes group of people called as Scrum Team. Normally It contains 7-12 Members.

1. Product Owner

2. Scrum Master

3. Development Team.

4. QA Team (Testing Team)

1. Product Owner: -

* Define the feature of the product.
* Prioritize features according to the market value.
* Adjust features and priority every iteration as needed.

2. Scrum Master: -

The main role is facilitating and driving the agile process.

3. Developer 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 the Product Owner.

The Requirement of Entire/Complete Application is called as Product Backlog.

**SPRINT/ ITERATION: -**

Period of time to complete the user stories, decided by the product owner and team.

Usually 2-4 weeks of time.

**SPRINT BACKLOG: -**

The Prioritized set of requirements which will be developed in a u

**SPRINT PLANNING MEETING: -**

Sprint Planning Meeting conducts with the team to define what can be delivered in the sprint and duration.

**SCRUM MEETING: -**

Meeting Conducted by Scrum Master (Everyday 15-30 Mins) called as Scrum Call/Stand Up Call/ Daily Standup Call

1. What you did Yesterday.

2. What you will do Today.

3. Are there any issues/impediments you facing.

**STORY POINT: -**

The Estimation of each and every module (No of days taken to develop, no of days to taken to test) is called as Story Point.

Rough Estimation of user stories will be given by Dev and Test team.

1 Story Point = 1 hour /1-day (6-7 hours)

**IMPEDMENTS BACKLOG**

As soon as Test Engineer talks about an obstacle let the Scrum master will prepare a document about all the obstacles that is called as Impediments Backlog.

**BURNDOWN CHART**

It Shows how much work remaining in the sprint.

Maintained by the ScrumMaster

**RETROSPECTIVE MEETING**

In this meeting, the entire Scrum Team meet and discuss the achievements and mistakes.

Good Practice that can be followed in the next Sprint. (Improvements)

**Smoke Testing and Sanity Testing:**

**Smoke Testing: -**

1. Smoke Testing is done to make sure that the build we received from development team is testable/stable or not.

2. Smoke Testing is performed by the developers and Testers.

3. Usually, it is done every time when there is new build release.

4. As a part of Smoke testing, we check whether the basic and critical features of an application before we do 'through' testing.

5. Smoke Testing is also known as Dry Run/ Build Verification Testing.

**Sanity Testing: -**

1. Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper.

2. Sanity Testing is performed by the Tester.

3. It is done on stable builds.

Registration: - Dev Env Code Migration > Testing Evn Testing

Retesting: -

Whenever the developer fixed a bug, tester will test the bug fix is called Retesting.

Tester closes the bug if worked otherwise re-open and send to the developer.

To ensure that the defect which are found and posted in the earlier build were fixed or not in the current build.

Example: -

Build Release V1.0. Test team should find some defect (Defect id 101,102,105)

Build V1.1 was released, now testing the defects (101,102,105) in this build is Retesting.

Test Plan: -

A Test Plan is a document that describe the test scope, test strategy, objective, schedule, deliverables and resources required to perform testing of a software product.

Test Plan Template Contents: -

- Overview - Scope \* Inclusion \* Test Environments \* Exclusion

- Test Strategy

- Defect Reporting Procedure

- Roles/ Responsibility

- Test Schedule

- Test Delivery

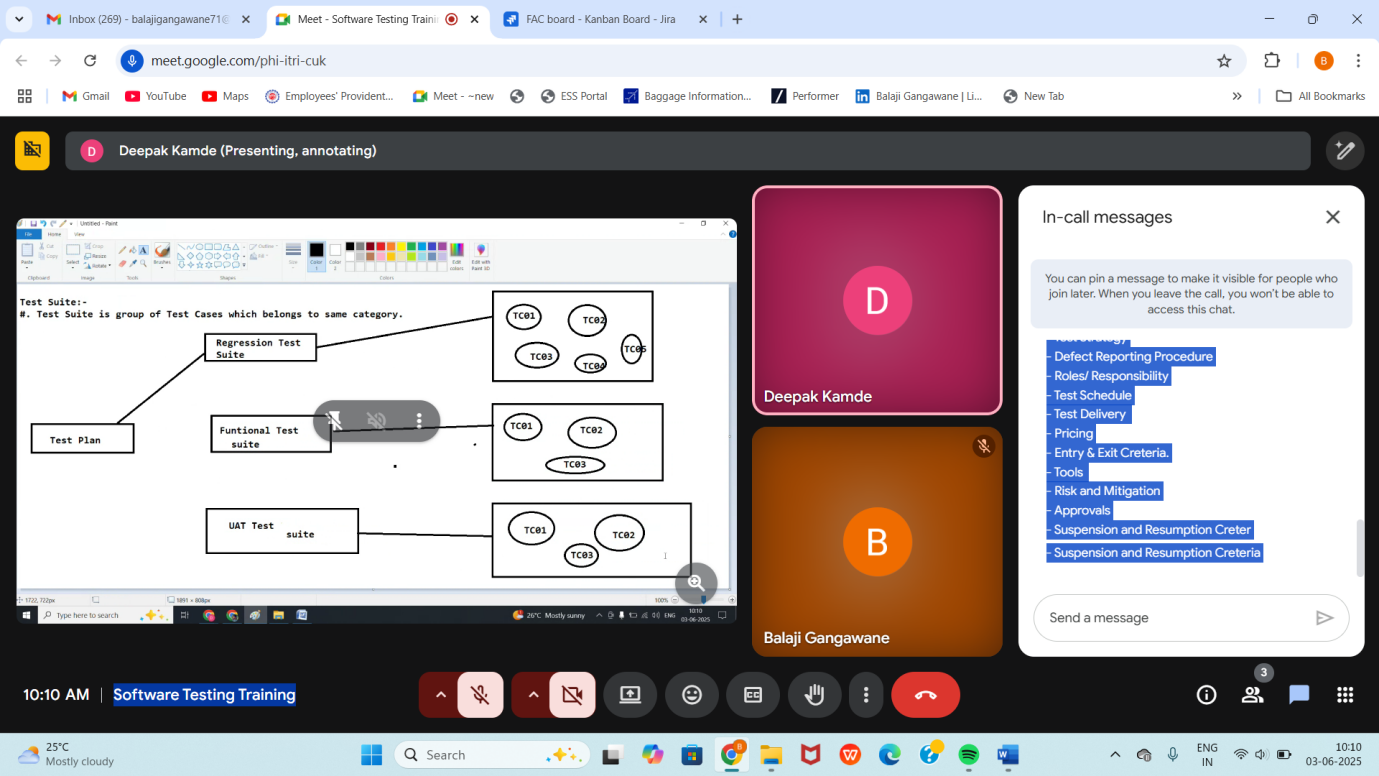
- Pricing

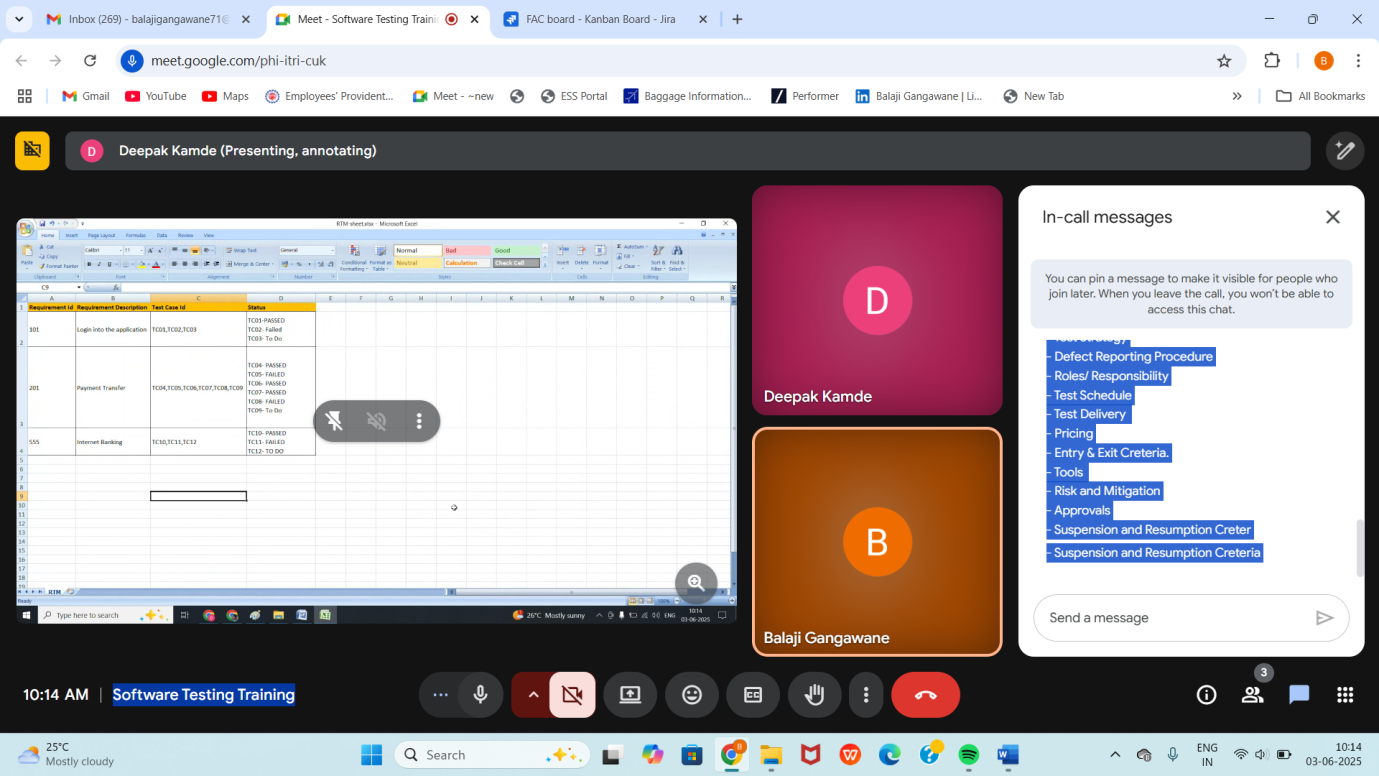
- Entry & Exit Creteria.

- Tools - Risk and Mitigation

- Approvals

- Suspension and Resumption Creter





Test Execution: -

During this phase test team will carry out the testing based on the test plan and test cases prepared.

Entry Criteria: - Test Environment, Test Cases, Test Data , Test Plan

Activities: -

#. Test Cases are executed based on the Test Planning.

#. Status of Test Cases are Marked as like 'PASSED' 'FAILED','BLOCKED', 'NOT APPLICABLE' and others.

#. Documentation of Test Result and log a defect for failed test cases is done. #. All the blocked and failed test cases are assigned bug ids.

#. ReTesting once the defects are fixed.

#. Defects are tracked till closure.

Deliverables: - Provide defect and test cases execution report with completed result.

Design Technique/ Test Case Design Technique / Test Design Technique: -

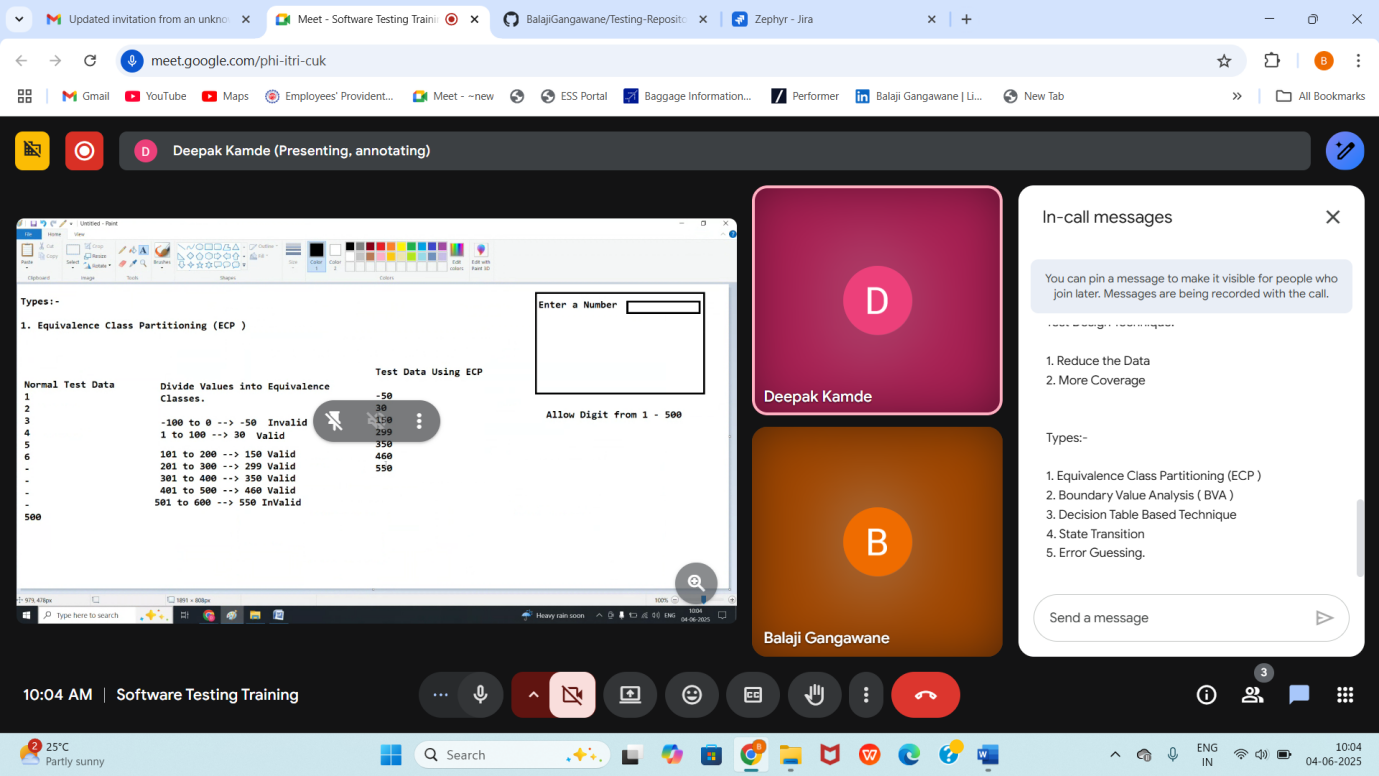
1. Reduce the Data

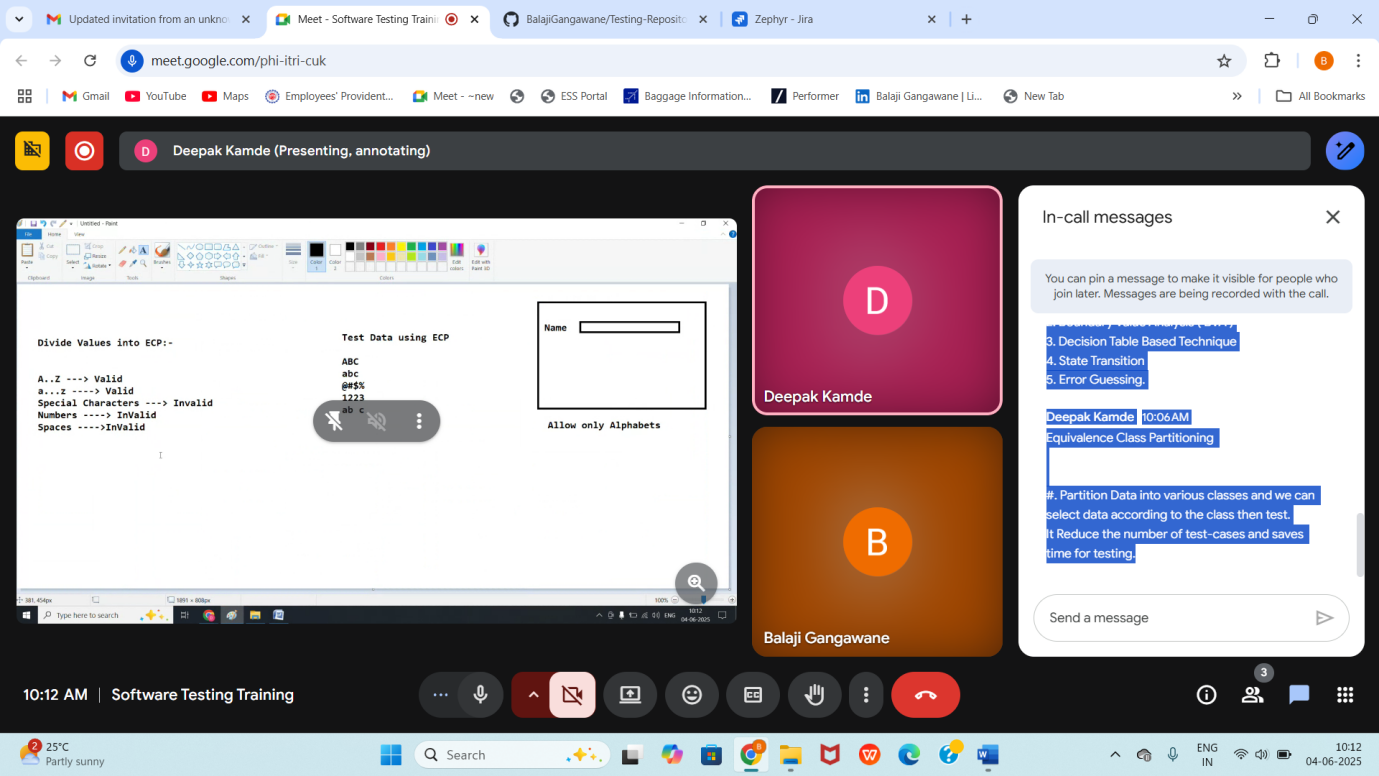
2. More Coverage

Types: - 1. Equivalence Class Partitioning (ECP ) 2. Boundary Value Analysis ( BVA ) 3. Decision Table Based Technique 4. State Transition 5. Error Guessing.

Equivalence Class Partitioning

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





Decision Table: -

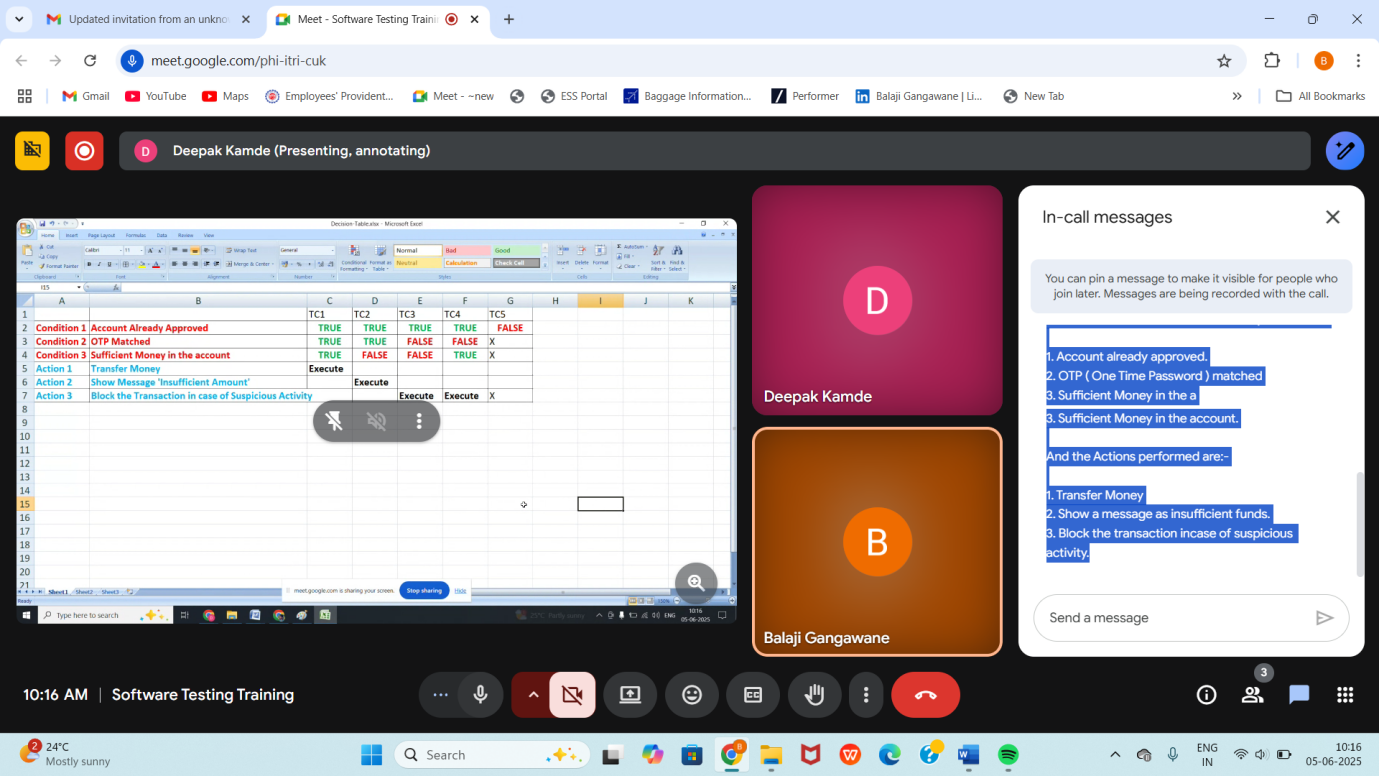
#. Decision Table is also called as Cause-Effect-Table.

#. This technique will be used if we have more conditions and corresponding actions.

#. In Decision Table Technique, we deal with combination of inputs.

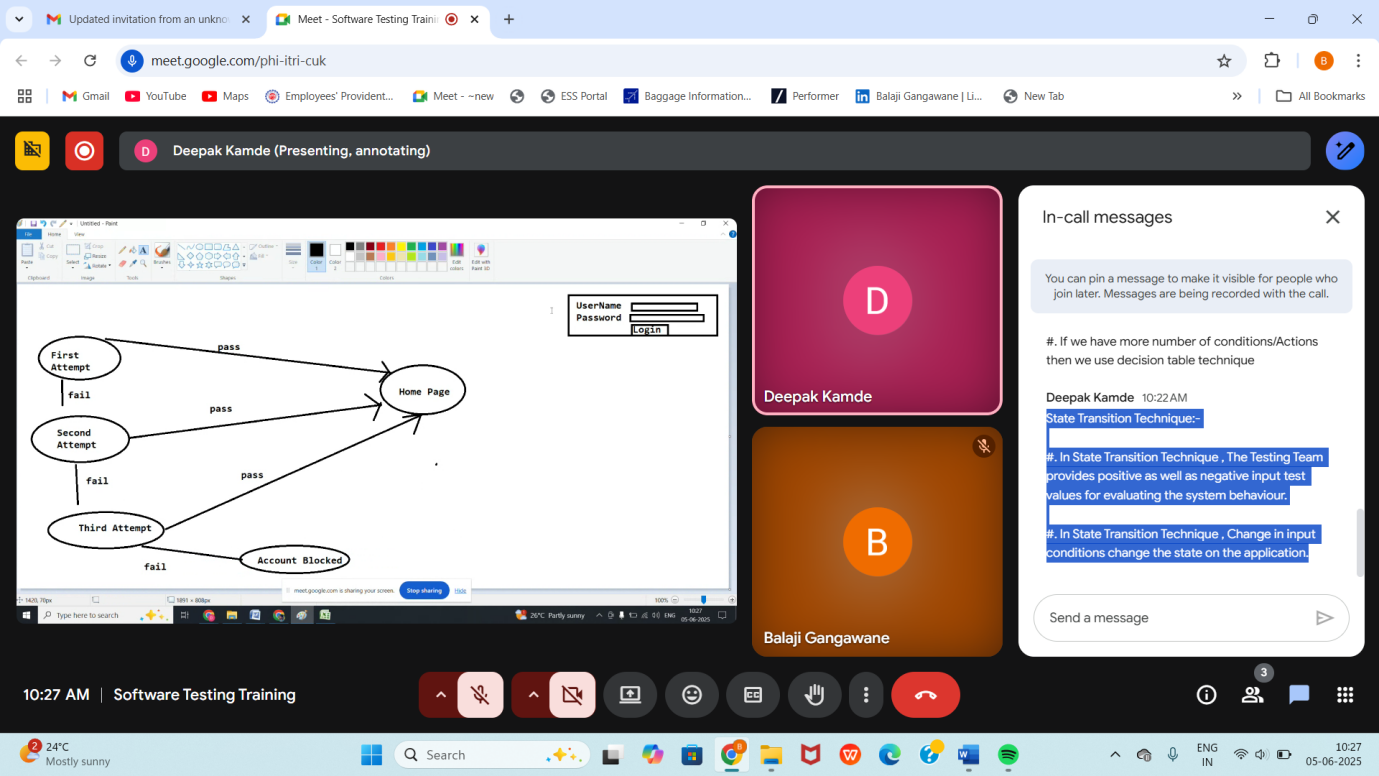
#. To Identify the test cases with decision table , we consider conditions and actions. Example:- Transfer Money Online:- Here, the conditions to transfer money online are:- 1. Account already approved. 2. OTP ( One Time Password ) matched 3. Sufficient Money in the a

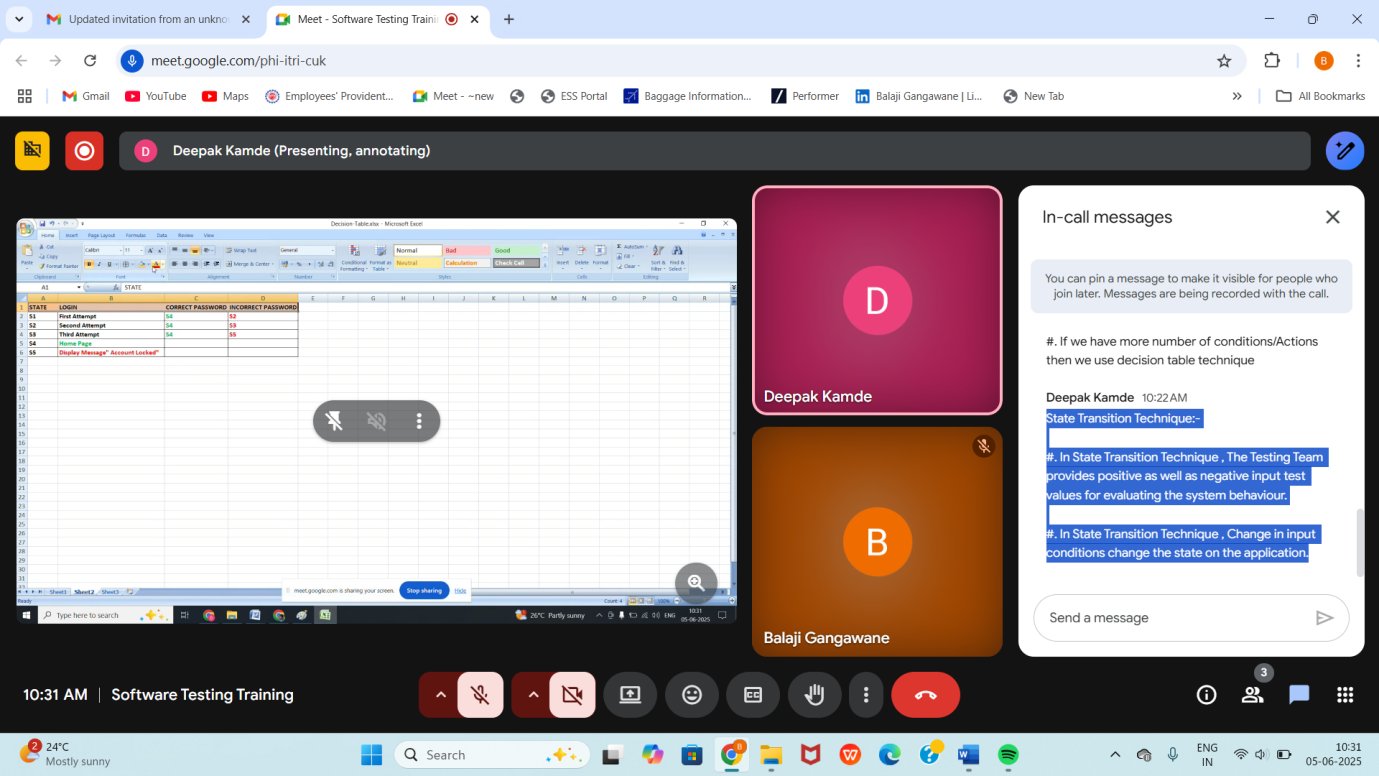
3. Sufficient Money in the account. And the Actions performed are:- 1. Transfer Money 2. Show a message as insufficient funds. 3. Block the transaction incase of suspicious activity.



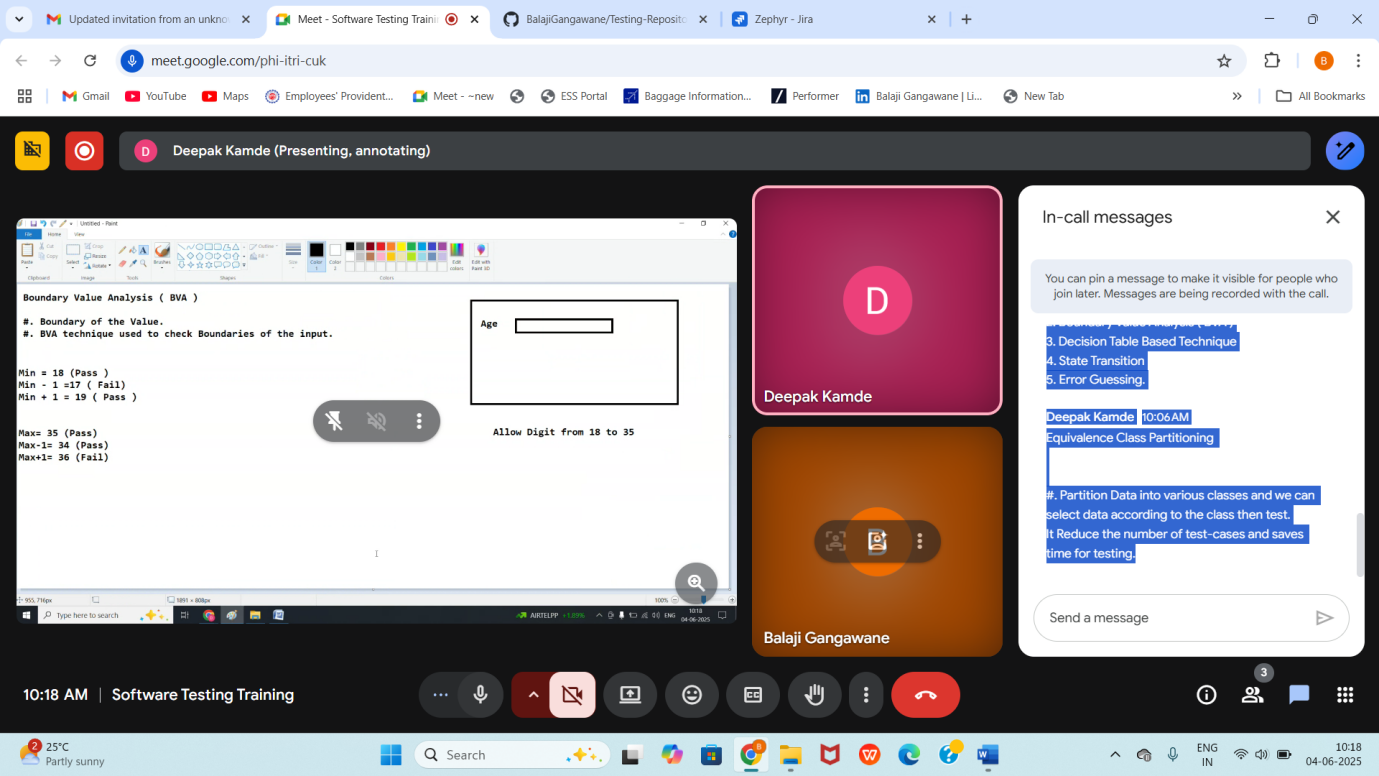
State Transition Technique:-

#. In State Transition Technique , The Testing Team provides positive as well as negative input test values for evaluating the system behaviour. #. In State Transition Technique , Change in input conditions change the state on the application.





Boundary Value Analysis (BVA)



100 to 199 Min =100 (Valid ) Min-1 =99(Invalid) Min+1=101(Valid) Max=199(Valid) Max-1=198(Valid) Max+1=200(Invalid)

Error Guessing: - #. Error Guessing is one of the testing technique used to find bugs in a software application based on testers prior experience. #. In Error Guessing we don't follow any specific rules. #. It depends on Testers Analytical Skills and Experience. #. Some of the Examples are - Submitting a form without entering values. - Entering Invalid Values such as entering alphabet in the numeric field.