**Integration Testing**

Testing the data flow between two or more modules is called as Integration Testing.

**Importance of Integration Testing**: Integration testing is a critical phase in the software testing process that focuses on verifying the interactions and interfaces between different components or systems within a software application.

**Integration testing is important because of following reasons:**

● It ensures proper data flow - Integration testing validates the proper flow of data between different modules and

● ensures that data is transferred accurately. Data consistency and accuracy are crucial

● for the correct functioning of the entire system. Integration testing helps uncover data-related issues that may arise when different components exchange information. ● Validating business workflows - Users interact with software applications through specific workflows. Integration testing ensures that these workflows, which often involve multiple components, are seamless and error-free.

● It helps to detect Interface issues - Integration testing helps identify issues related to the interaction between different modules, components, or services.

● It ensures System Stability - By verifying the integration of components, integration testing helps ensure that the system operates cohesively and reliably, meeting performance and stability requirements.

**There are 2 types of Integration Testing:**

a) Incremental Integration Testing

b) Non-incremental Integration Testing

a) **Incremental Integration Testing –** Incrementally adding the modules and testing the data flow between the modules is called Incremental

Integration Testing.

It is of 2 types: **Top Down approach and Bottom-up approach**

1) **Top Down approach** – Incrementally adding the modules and testing the data flow between the modules and ensuring that the modules we are adding are the child of the previous module is called the Top Down

approach. Here data will flow from top to bottom.

2) **Bottom-up approach** – Incrementally adding the modules and testing the

data flow between the modules and ensuring that the modules we are adding are the parent of the previous module is called the Bottom-up

approach. Here data will flow from bottom to top.

b) **Non-Incremental Integration Testing** – Here we randomly test the data flow between all other modules. We go for this testing, when we don’t

know which parent and child module or when requirements are

complex to understand, we can do Non-Incremental Integration Testing. **Positive Integration Testing scenario on Gmail:**

● Login as a user, click on compose. Enter some information, now click on cancel. Now click on the draft option, check that the information

written so far should be displayed.

**Negative Integration Testing scenario on Gmail:**

Login as user, click on compose. Enter some information and click on

cancel. Click on Trash and check if the entered details are displayed.

**System Testing**

**It** is an end to end testing conducted by the Test engineers in a testing environment which is similar to a production environment.

**What is End to end testing?**

Navigating through all the features and checking whether the end feature is working as expected or not is called end to end testing.

End-to-end testing is a comprehensive software testing approach that evaluates the entire software application from start to finish, including all integrated components and external dependencies. The primary goal of end-to-end testing is to ensure that the entire system functions as expected and meets the specified business requirements. This type of testing simulates real-world scenarios to validate the software's behavior in a production-like environment.

In this testing, we test all the end-to-end features and check whether it is according to the customer requirements. It is very essential to do end to end testing because customers in real time do verify end to end features of an application and if it is working fine or not.

**System testing is a critical phase in the software testing process that evaluates the entire software system as a whole. It is conducted after integration testing and**

**before**

**acceptance testing, focusing on verifying that the integrated system functions according to the specified requirements.**

**Importance of System testing:**

● System testing verifies the entire system including all integrated components and functions as intended.

● System testing identifies all System level defects, that may surface only when complete system functionalities are assembled.

● It verifies all individual components when integrated collectively fulfill the intended business and functional requirements.

● System testing ensures end to end business scenarios are met for better user experience.

**Positive System testing scenario on Flipkart:**

● Log in to flipkart, click on mobiles feature, click on samsung, select one of the samsung mobile. Now click on buy now, select the payment option as phonepe, complete the payment. Click on the orders and check the product details are displayed.

**Negative System testing scenario on Flipkart:**

● Login to flipkart, search for fastrack watches, select one of the watches. Click on buy now, click on continue, select the payment option as debit card. Generate one time password. Enter the wrong OTP and check if error message is displayed.