

**Department: CS&SE Class: Bse-6D**

**Class Assignment No.01 Subject: software testing**

**Total Marks: 10 submission date: 20-04-22**

**Name: Rohail Khan**

**Reg no: FA19-BSE-131**

**Q. No. 01**

**Describe the main challenges in integration testing? And please find the good suggestions to overcome on these challenges?**

**What is integration testing?**

The meaning of Integration testing is quite straightforward- Integrate/combine the unit tested module one by one and test the behavior as a combined unit.

The main function or goal of this testing is to test the interfaces between the units/modules.

We normally do Integration testing after “Unit testing”. Once all the individual units are created and tested, we start combining those “Unit Tested” modules and start doing the integrated testing.

The main function or goal of this testing is to test the interfaces between the units/modules.

The individual modules are first tested in isolation. Once the modules are unit tested, they are integrated one by one, till all the modules are integrated, to check the combinational behavior, and validate whether the requirements are implemented correctly or not.

Here we should understand that Integration testing does not happen at the end of the cycle, rather it is conducted simultaneously with the development. So in most of the times, all the modules are not actually available to test and here is what the challenge comes to test something which does not exist!

**Why Integration Test?**

We feel that Integration testing is complex and requires some development and logical skill. That’s true! Then what is the purpose of integrating this testing into our testing strategy?

**Here are some reasons:**

1. In the real world, when applications are developed, it is broken down into smaller modules and individual developers are assigned 1 module. The logic implemented by one developer is quite different than another developer, so it becomes important to check whether the logic implemented by a developer is as per the expectations and rendering the correct value in accordance with the prescribed standards.
2. Many a time the face or the structure of data changes when it travels from one module to another. Some values are appended or removed, which causes issues in the later modules.
3. Modules also interact with some third party tools or APIs which also need to be tested that the data accepted by that API / tool is correct and that the response generated is also as expected.
4. A very common problem in testing – Frequent requirement change! :) Many a time developer deploys the changes without unit testing it. Integration testing becomes important at that time.

**Advantages**

**There are several advantages of this testing and few of them are listed below.**

* This testing makes sure that the integrated modules/components work properly.
* Integration testing can be started once the modules to be tested are available. It does not require the other module to be completed for testing to be done, as Stubs and Drivers can be used for the same.
* It detects the errors related to the interface.

**Challenges**

**Listed below are few challenges that are involved in Integration Test.**

**#1)** Integration testing means testing two or more integrated systems in order to ensure that the system works properly. Not only the integration links should be tested but an exhaustive testing considering the environment should be done to ensure that the integrated system works properly.

There might be different paths and permutations which can be applied to test the integrated system.

**#2)** Managing Integration testing becomes complex because of few factors involved in it like the database, Platform, environment etc.

**#3)** While integrating any new system with the legacy system, it requires a lot of changes and testing efforts. Same applies while integrating any two legacy systems.

**#4)** Integrating two different systems developed by two different companies is a big challenge as for how one of the systems will impact the other system if any changes are done in any one of the systems is not sure.

In order to minimize the impact while developing a system, few things should be taken into consideration like possible integration with other systems, etc.

**Suggestions to overcome on these challenges?**

Ideally, your collaboration product delivers efficiency, productivity, and flexibility to its users. Applying a relevant third-party app integration - single sign-on capability, social media sharing features, customer relationship management, and document editors - adds considerable value to your product and greatly improves the product’s user experience. And thorough 3rd party QA services ensure that each 3rd party integration works as expected and does not affect the existing functionalities of your collaboration product.

Thoughtful integration of third-party applications can make your collaboration product more beneficial and more appealing to your users. That being said, even the most effective third party software testing solutions can run into these three integration challenges:

1. 3rd party app integration errors disguised as product errors.
2. Access issues between third party system integration and your product.
3. Incompatibility of the third party integration and your product interface.

Simply put, the best way to tackle these challenges is by having solid QA testing procedures in place so that your team is equipped with the right skills and habits in overcoming any obstacle with third party integrations for your collaboration product.

Before we take a deeper diver on how to integrate third party API successfully with your software application, let’s first explore the value third party integrations bring to a collaboration product.