# Why Integration Testing?

Main Research Question:

* Why to use Integration Testing?

Sub Research Questions:

1. What is integration testing?
2. What is unit testing?
3. What is End to End testing?
4. Why to choose integration testing over unit testing?
5. Why to choose integration testing over end-to-end testing?

Answers To The Sub Research Questions

1. What is integration testing?

Integration Testing, for example, helps in testing the connection to a database and check if you are able to query the database the usually would in normal usage, also sometimes bugs appear during usage that don’t appear during normal tests such as unit testing, where it finds issues that are not as obvious when you use unit testing or by examining the application.

Integration Testing discovers defects in the interplay of several application parts. Sometimes, these defects can be challenging to track or reproduce, Integration testing will aggregate results from various parts and sources.

You do not need to mock parts of the application with integration testing, you can replace external systems, but the application works in an integrated way.

2. What is unit testing?

Unit testing is primarily used for focusing on a specific part of an entire application, usually a function or just a single class, also the issue with unit testing is the ability to only public functions rather than private ones.

However, unit testing does have the ability to quickly perform the required tests in speeds much shorter than that of integration testing, it is also essential in test driven development.

The specifications of unit tests must be exact to be able to be exact for them to be able to be carried out in the proper way, also for the continuation of code implementation and test writing which does not comply with the Agile dimensions.

3. What is E2E testing?

End to End testing is the method of testing that involves testing an applications workflow from beginning to end, it goes through every small part of every function that the application has, and how it communicates with the hardware, network connectivity, external dependencies, and databases.

This method of testing basically aims to replicate real user scenarios so that the system can be validated for integration and data integrity.

4.Why to choose integration testing over unit testing?

Basically you do not need to fully test all the end points of an api by creating mocks and focusing on one specific functions, it does not discover bugs that come with normal user usage in the way integration testing does, it is also not as effective when it comes to testing CRUD functions, furthermore integration testing is easier to implement by a developer who is unfamiliar with the source code and thus it will comply with agile.

5.Why to choose integration testing over end-to-end testing?

The end to end testing method is not suitable for my application as it goes through every function that the program has, which is ineffective due to its need to the needlessness of going through every function in a crud application where only some end points are important to test over others.