

Software Testing Analysis

Team 38

Project 4 - Quantification of CFP for Logistic Hubs

One of the most important stages in the software development life cycle is the 'Testing' phase. This is where the developers check whether the code is working according to the requirements mentioned in the SRS. There are different types of software testing methods we used at different levels. Different levels of testing are **UNIT TEST** , **INTEGRATION TEST** , **SYSTEM TEST AND ACCEPTANCE TEST**.

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. When the complete pen is integrated, System Testing is performed. Once System Testing is complete, Acceptance Testing is performed so as to confirm that the ballpoint pen is ready to be made available to the end-users.

1) Acceptance Testing

ACCEPTANCE TESTING is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. It is done after the unit testing, integration testing and system testing.

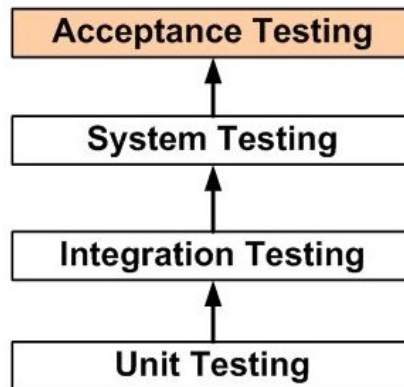
It is a formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Black Box Testing method is used in Acceptance Testing. **BLACK BOX TESTING**, also known as Behavioral Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester.

Our Implementation of Acceptance Testing:

Once the whole project and the system testing is done, we have realised that we have restricted ourselves to some domain of tests as we know the functionalities and what all we have implemented. So, we have decided to make some other person (not the developer of the project) to perform the tests. And, we did that. By doing that, tests can be done from a user's point of view and will help in exposing discrepancies in the specifications. After the test, we got to know some of the error handlings that users expect and we solved them.

Then we have shown it to the client to know if the app has met their expectations and requirements.

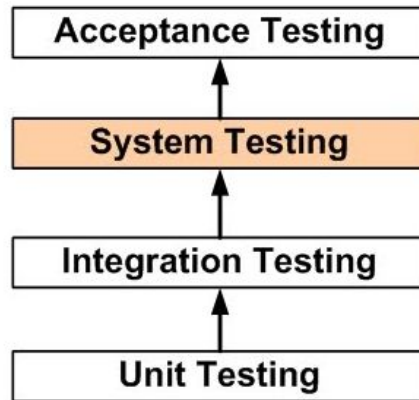


2) System testing

SYSTEM TESTING is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. It is tested before the acceptance testing after the whole system is integrated and integration and unit testing are done.

Our Implementation of System Testing:

We performed this after we are done with the project. All 4 of us tested the whole system using white box testing and black box testing. We have tested and found out whether the whole system is working by testing all the modules and features implemented.



3) Integration testing

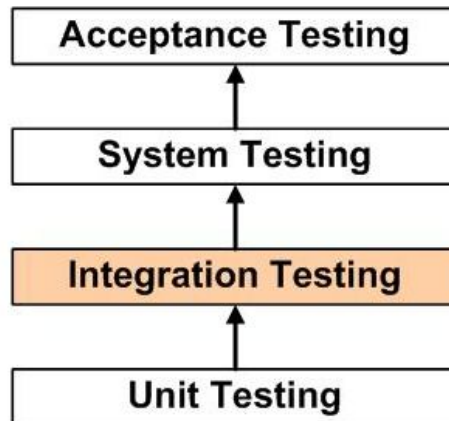
INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Performed after unit testing.

- **Big Bang** is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle. It tests the interfaces only that makes it different from the system testing
- **Top Down** is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when a top-down development approach is followed.
- **Bottom Up** is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when a bottom-up development approach is followed.
- **Sandwich/Hybrid** is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

Our Implementation of Integration Testing:

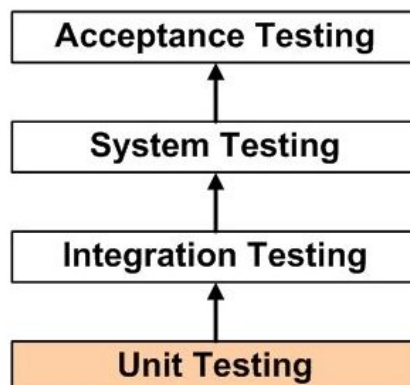
We used the top-down approach of integration testing. We have tested the high-level modules first and then the low-level modules and finally integrating the low-level

modules to the high level to ensure the system is working as intended. Considering the whole display of maps and open layers part as a high level module and CFP calculation and login module as low level modules. Both of those modules are independent and the only interface is the username that we hardcoded during the unit test. So, testing high level modules first helped us to make sure that the major module is working. And integrating it with low-level modules was then easy.



4) Unit testing

UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. It is the first level of testing.



Units in our product are the modules most of which we implemented by R1 Code Freeze:

- 1) Login module
- 2) CFP calculation module (Query part)
- 3) Maps module

WHITE BOX TESTING is a software testing method in which the internal structure/design/implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.

Our Implementation of White Box Testing:

We have used white box testing for unit tests. For example, for the login module, the testing goes like this. The user should first register. We test if all the error-handlings are done. Whenever there is an error, the user can't register and the user can't be found in the DB. To get to know the path and working we have used console.log statements in the frontend and print statements in the back-end. If there is no error, then the user can be registered and the entry will be found in DB. Each time checking the console, terminal and DB for print statements and entries made it easy to know if it's functioning well or not.