

Software Testing Analysis

Following are the levels of testing. On each level, the system components were tested either as a part or whole.

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. Acceptance Testing is the last level of software testing, after System Testing.

For the final product, the black box testing method is used.

Black Box Testing Method: Testing, either functional or non-functional, without reference to the internal structure of the component or system. The test cases are derived from the functional and non-functional specifications of the product.

At this level of testing, the product must be proven deployable and to attain an objective approval, an end-user/customer evaluates various test cases per the specifications in the SRS. Our product was tested by known neutral parties without any knowledge of the source code.

Examples:

1. Some NULL exceptions were not taken care of when searching for startups. E.g., the region/country attribute was left empty or a non-existent profile link (NULL value) was displayed.

Expected Outcome: Profile is not displayed.

Observer Outcome: Profile button redirects to a blank page.

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. System Testing is the third level of software testing, after integration testing.

It is similar to Acceptance Testing regarding the test setting and methodology. Both use the black box testing method to verify if the software system as a whole satisfies the specified requirements. However, system testing is done by the testers in the production teams as opposed to end-users/customers in Acceptance Testing.

Examples:

1. In certain test cases, some UI elements were misplaced.

Expected Outcome: The UX is consistent.

Observed Outcome: The information was difficult to infer.

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. Integration Testing is the second level of software testing, after Unit Testing.

There are four standard approaches to Integration Testing:

1. Bottom Up Integration:

Bottom Up is an approach to Integration Testing where bottom level units are tested first, and upper-level units are tested step by step after that. This approach is taken when bottom-up development approach is followed.

2. Top Down Integration

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 top-down development approach is followed.

3. Big Bang Integration

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 is different from System Testing as it only tests the interaction between the components/units of the system.

4. Sandwich Integration

Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

The product was developed with a Bottom Up approach using Agile methodology. Hence, the Bottom Up approach to Integration Testing was suitable for our product.

The product initially started with develop the units individually without any pre-defined interactions. Therefore, the Bottom Up approach seemed feasible in the initial stages of the development. Each unit was tested thoroughly and its interactions with others, after that.

Examples:

1. The SSO login unit was developed first alongside the search feature which did not require a user to be logged in. Once each unit was verified working, both were integrated and tested if the searches were user specific. Initially, a search without a user logged in would be stored causing NULL exceptions.

Expected Outcome: Search failed due to unauthenticated user.

Observed Outcome; Search proceeds with empty results.

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 software testing, before Integration Testing.

Unit Testing is done using the White Box Testing method.

White Box Testing Method: Testing based on an analysis of the internal structure of the component or system. The test cases are derived by observing the internal structure of a component/unit of a system.

Examples:

1. The domain attribute was supplied other than the allowed list of domains for the search form unit.

Expected Outcome: Search failed due to incorrect input, input again.

Observed Outcome: Search successful, results empty.