7/23/2018

Testing Methodology:

* Make sure software meets its specification/requirement and can successfully operate in all the anticipated environments with the required usability and security.
* Has no undesirable side effect when used in ways outside of its design parameter
* And worst case will fail-safely

Gregory Masters, 408-236-6400

**Functional Testing**:

1. Unit testing

* Test of individual software modules or components that make up an application or system. Ensure full coverage of all program code and all unit tests have passed.

1. Integration Test

* Testing of different modules/components when integrated together to perform specific tasks and activities. All the modules have been unit – tested.

1. System Testing

* Testing the entire system for errors and bugs. The test is carried out by interfacing the hardware and software components of the entire system (that have been previously unit tested and integration tested and now testing it as a whole.
* This testing is listed under the black-box testing method, where the software is checked for user-expected working conditions as well as potential exception and edge conditions.

1. Acceptance Testing

* final phase of functional software testing and involves making sure that all the product/project requirements have been met and that the end-users and customers have tested the system to make sure it operates as expected and meets all their defined requirements:

//

Functional Testing:

* Smoke test, Sanity test
* Integration test, system test
* **Regression test**, User acceptance test
* Provide input, get output, and compare
* The key is not functional testing, but write functions to test.
* Focus on how to write function to test.

**Non-functional Testing**

1. **Performance testing** is measuring how a system behaves **under an increasing load** (both number of **users**/DUTs and **data volumes**), load testing is verifying the system can operate at the required **response times** when subjected to its expected load, and **stress testing** id finding the failure points in the system when the tested load exceeds what it can support.
2. Performance ( increase DUT from single DUT to 8 or 16 DUTs.)

* Increase # of DUTs from 1 up to 16.
* Increase test items to a certain number
* Increase # of test runs.
* How about virtual DUTs, virtual testers, virtual clients.

1. Improvement in

* Test time
* Test coverage, corner cases, exception.

1. **Security**, Vulnerability Testing
2. **Usability Testing**

**Stress Testing**

* Execution of application above the threshold of the break to determine the point where the application crashes. The condition usually arises when there are too many users and too much of data.

**Load Testing**

* Application is executed above various load levels to monitor performance of the server, **response time**, server throughput and determine stability, performance and integrity of the application.

Test of login feature of any application:

* Check the input field, username, passwd with both valid and invalid values.
* Check the proper error message displayed.
* Enter valid credentials and get logged in. Close and reopen the browser to check if it still logged on.
* Enter the application after logging in, and then again navigate back to the login page to check if the user is asked again to login or not.
* Change passwd , and try to log in with the old passwd.

**Regression Testing**

* Make sure the implementation of any new feature or fixes does not affect any other part or functionality of the application.
* New changes in the application should not affect the existing functionality
* As manual testing can be time consuming and expensive, automation can be done for regression testing.
* LifeCycle of an application testing

1. Design test plans based on application requirement
2. Develop manual test cases and automated test scripts
3. Execute functional tests to validate application requirement
4. Execute load tests and tune application performance

* Application Testing Methodology

1. Black box
2. White Box
3. Grey Box