**Test Execution – How to Deal with this Most Important and Happening Phase in the STLC – QA Training on a Live Project Day 5**

Posted In | [Software Testing Training](http://www.softwaretestinghelp.com/category/software-testing-training/), [Testing Life cycle](http://www.softwaretestinghelp.com/category/testing-life-cycle/)

Today, in our**software testing *mini*training course**, we are progressing into the last stage of the STLC, which is the ***test execution***.

You can check out the list of all tutorials posted in this free QA training series on this page: [End to End software testing training on a live project.](http://www.softwaretestinghelp.com/free-online-software-testing-qa-training-course/" \o "Software testing training mini course)

**Test execution is without doubt the most important and ‘happening’ phase in the [STLC](http://www.softwaretestinghelp.com/what-is-software-testing-life-cycle-stlc/" \o "STLC) and also the entire development life cycle.** The reason is – every team/team member’s contribution and work gets validated here:

* Has the Business Analyst interpreted the requirements correctly?
* Has the development team translated the business requirements to functional requirement and eventually to code correctly?
* Has the data architect and DBAs design the right back end systems?

Well, test execution is where all the answers to these questions would be found. That does make us, QAs the heroes of the entire software building process, doesn’t it? :)

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/test-execution-STLC.jpg)

**Test Execution is also the “Test” part of the SDLC.**

Once the test cases are written, shared with the BAs and Dev team, reviewed by them, changes are notified to the QA team (if any), QA team makes necessary amends- Test design phase is complete. Now getting the Test cases ready does not mean we can initiate the test run. We need to have the application ready as well among other things.

**Test Execution Guidelines:**

**Let us now make a list of all things that are important to understand Test Execution phase:**

**#1. [The build](http://www.softwaretestinghelp.com/bvt-build-verification-testing-process/" \o "Build verification process)** (the code that is written by the dev team is packaged into what is referred to a build- this is nothing but an installable piece of software (AUT), ready to be deployed to QA environment.) being deployed (in other words, installed and made available) to the QA environment is one of the most important aspects that needs to happen for the test execution to start.

**#2.** Test execution happens in the **QA environment**. To make sure that the dev team’s work on code is not in the same place, where the QA team is testing, the general practice is to have dedicated Dev and QA environment. (There is also a production environment to host the live application). This is basically to preserve the integrity of the application at various stages in the SDLC life cycle. Otherwise, ideally, all the 3 environments are identical in nature.

**#3.** **Test team size** is a not constant from the beginning of the project. When the test plan is initiated the team might just have a Team lead. During the test design phase, a few testers come on board.  Test execution is the phase when the team is at its maximum size.

**#4.** Test execution also happens in **at least 2 cycles** (3 in some projects). Typically in each cycle, all the test cases (the entire test suite) will be executed. The objective of the first cycle is to identify any blocking, critical defects, and most of the high defects. The objective of the second cycle is to identify remaining high and medium defects, correct gaps in the scripts and obtain results.

**#5.** Test execution phase consists of- **Executing the test scripts + test script maintenance (correct gaps in the scripts) + reporting (defects, status, metrics, etc.)** Therefore, when planning this phase schedules and [efforts should be estimated](http://www.softwaretestinghelp.com/software-test-estimation-how-to-estimate-testing-time-accurately/" \o "Test estimation) taking into consideration all these aspects and not just the script execution.

**#6.** After the test script being done and the AUT is deployed – and before the test execution begins, there is an intermediary step. This is called the **“Test Readiness Review (TRR)”**.  This is a sort of transitional step that will end the test designing phase and ease us into the test execution.

For information on this step and a sample “Test readiness review checklist”, check out this link: [Software testing Checklist](http://www.softwaretestinghelp.com/software-testing-qa-checklists/" \o "Software testing qa checklist)

**#7.** In addition to the TRR, there are few more additional checks before we ensure that we can go ahead with accepting the current build that is deployed in the QA environment for test execution.

Those are the **smoke and sanity tests**. Detailed information on what these are is at: [What is Smoke and Sanity Test?](http://www.softwaretestinghelp.com/smoke-testing-and-sanity-testing-difference/" \o "Smoke test Sanity test)

**#8.** On the successful completion of TRR, smoke and sanity tests, the test cycle officially begins.

**#9.** **[Exploratory Testing](http://www.softwaretestinghelp.com/exploratory-testing-beyond-traditional-testing-boundaries/" \o "Exploratory testing)** would be carried out once the build is ready for testing. The purpose of this test is to make sure critical defects are removed before the next levels of testing can start. This exploratory testing is carried out in the application without any test scripts and documentation. It also helps in getting familiar with the AUT.

**#10.** Just like with the other phases of the STLC, work is divided among team members in the test execution phase also. The division might be based on module wise or test case count wise or anything else that might make sense.

**#11.** The primary outcome of the test execution phase is in the form of reports – primarily, defect report and test execution status report. The detailed process for reporting can be found at: [Test executions reports](http://www.softwaretestinghelp.com/test-execution-report/" \o "How to report test execution)

**New Columns in Test Cases Document:**

The test case document now gets to be expanded with the following two columns – **Status and Actual result**.

*(****Note****– For live project test execution, we have added and updated these columns with test execution results in the test cases spreadsheet provided for download below)*

------------

**Status column:**

Test execution is nothing but, using the test steps on the AUT, supplying the test data(as identified in the test case document) and observing the behavior of the AUT to see if it satisfies the expected result or not. If the expected result is not met, it can be construed as a defect. And the status of the test case becomes “Fail” and if the expected result is met, the status is “Pass”. If the test case cannot be executed because of any reasons (an existing defect or environment not supporting) the status would be “Blocked”. The status of a test case that is yet to be run can be set to No run/unexecuted or can be left empty.

* **For a test case with multiple steps, if a certain step’s (in the middle of the test case steps) expected result is not met, the test case status can be set to “Fail” right there and the next steps need not be executed.**
* **The status “Fail” can be indicated in red color, if you would like to draw attention to it immediately.**

**Actual result column:**

This is a space where we testers can record what the deviation in the expected result is. When the expected result is met (or a test case whose status is “Pass”) this field can be left empty. Because, if the expected result is met it means the actual result=expected result, which means rewriting it in the actual result column will be a repetition and redundancy.

**A screenshot of the deviation can be attached in this column for enhanced clarity of what the problem is.**

**Test Execution Results for OrangeHRM Live Project:**

Let us now get OrangeHRM and carry out the test execution based on the above guidelines listed. Here are a few points to note:

* The extended test case template.
* Exploratory testing as indicated is to be carried out without test scripts. So please feel free to test the application in parallel as you see fit.
* Due to the limitations that we have in presenting the live project in the form of readable content- only a limited amount of test cases/functionality of the OrangeHRM application is shown in the sample test execution template. Again, please feel to work on more for the most practical experience.
* The sanity and smoke test suites are also added to the document, to give you an idea about what kind of test cases are considered for these stages.
* Defects are not logged yet, even though the status of some test cases is set to “Fail”. This is because, logging the defects is the next most important/commonly worked on aspect of our life as testers. So, we want to deal with it in detail in the next article.

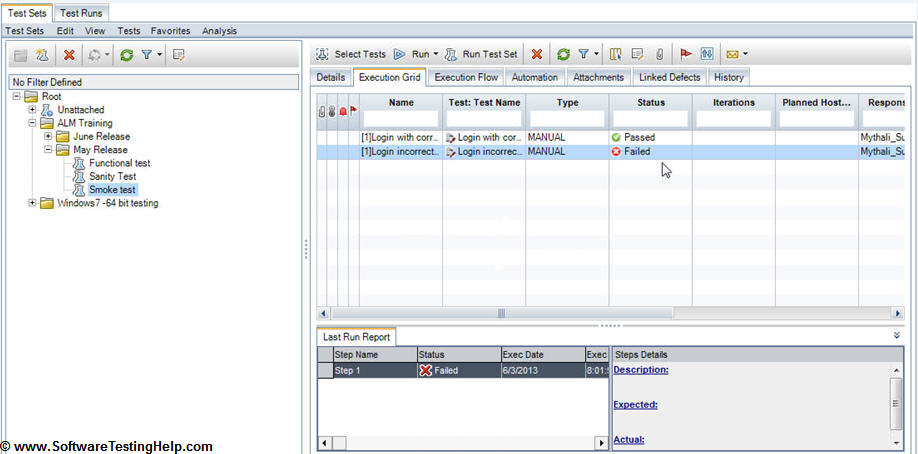
**Test Cases with Execution Results:**

=> **[Click here to download the test case execution document.](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/Test-execution-result-for-OrangeHRM-Test-Cases.xlsx" \o "Test execution result " \t "_blank)**

***It Contains****– Test cases execution result, smoke tests, sanity tests, exploratory test – spreadsheets*

Lastly, if a test management tool was used for creating and maintaining the test case, the same can be used for test execution as well. The use of a tool makes reporting easier, but otherwise the process of running the test cases is the same.  Please check out this article to get an idea on [how to use HP ALM for test case execution](http://www.softwaretestinghelp.com/hp-quality-center-tutorial-5/" \o "Test execution using HP Quality Center).

***(Click on image to enlarge)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/02/ALM-for-test-excution-result.jpg)

This brings us to the end of another interesting segment of the testing process. In the next and last article of this **free online Software Testing QA training mini course**, we will look into defects in detail; wrap up topics like “when to stop testing”, metrics and QA sign off.

**=> QA Training Day 6: [Bug Tracking, Test Metrics, and Test Sign off](http://www.softwaretestinghelp.com/bug-tracking-test-metrics-and-test-sign-off-free-qa-training-day-6/" \o "QA Sign off)**