10/23/2021



|  |
| --- |
| **Functional Testing Best Practices** | Pyramid Tech Riders |

|  |  |
| --- | --- |
| **Pyramid Solutions** | **Functional Testing Best Practices** |

Functional Testing Best Practices

**Introduction:**

Quality is a critical factor for software projects/products and customers. Now a day’s customers have become more tech-savvy and sophisticated and software business has become extremely competitive. Projects are behind scheduling because of multiple factors that include requirements changing so rapidly. The quality of project/product is determined by, besides other factor like skill levels of team members, the quality of software processes that produced it and the effectiveness of software testing is determined by the quality of testing processes used.

Agile has somewhat blurred the line of development and QA engineers. With focus on completing Product Back Log item/User Story in a sprint fashion, any team members can take up QA role for the particular “Testing” tasks if he/she has nothing else to do. Thus focus is on completing the User Story and getting feedback from the customer early.

**Need for Functional Testing:**

Functional testing consists of testing the interface between the application on one side and the rest of the system and users on the other side. It is a means of ensuring that project/product works as it should and what users expect it to do. Traditionally, functional testing is performed by a team of Quality Analyst. In Functional testing, QA capture requirements and provide confidence to developers and users that under testing project/product meets requirements and to confirm that User Story/Project/product is ready for production release.

**Functional Testing – Life Cycle, Challenges & Best Practices**

Testing requires a planned approach which includes test planning, test strategy, test designing, test execution and analysis to ensure maximum test coverage, consistency and re-usability of testing efforts.

Functional testing starts with gathering the testing requirements and continues through designing and developing tests, executing those tests and analyzing product defects. Various testing process steps can overlap and obviously it differs depending on each organization’s practices and methodologies. E.g. in Scrum based Agile process, you may not raise any defect during a Sprint (Iteration). But once User Story is claimed as “Done”, any defects found later on should be tracked as Defects. The fundamental principles of every testing process, however, remain the same. The fundamental aim of any test manager is to have an effective and efficient method for organizing, prioritizing and analyzing.

Quality depends on the effective planning and execution for the various stages in functional testing life cycle.

Let us take a look at various testing efforts at different stages of testing cycle with their challenges and suggested best practices.

**Stages of testing cycle:**

1. Test Requirements Gathering
2. Test Planning
3. Test Strategy
4. Test Execution
5. Defect Management
6. Test Results Reporting
7. Test Metrics Collection
8. **Test Requirements Gathering**

**Challenges:**

* Test requirements – Define clear and complete test requirements
* Scope creep – Manage changes to requirements

**Best Practice:**

* Arrange for product feature presentations from business team. This helps in minimizing communication gap between market requirements and what team actually develops. In Agile, this is one of key principle to demo every iteration to ensure requirements are well understood and implemented as business team/Product Manger wants.
* Prepare traceability matrix which show the relationship from features to requirements and requirements to test cases and test cases to defects.
* Maintain traceability matrices for any requirement changes; make corresponding modifications to test cases, test plans and traceability matrices at both levels.
* QA should collaborate closely with development team and business analyst throughout the testing life cycle starting with test requirements gathering stage and going upto the product release.
* Reduction in learning curve for each feature by the QA as the traceability matrices are quick and easy to go through.
* If QA is part of requirement elicitation, he/she can understand feature expectations and also provides feedback thus making him/her engaged in project/product from beginning.

1. **Test Planning**

**Challenges:**

* Gaps in test plan for functional testing.
* Difficulty in understanding the test plan for “To Test” and “Not to Test” requirements.

**Best Practice:**

* Have approved /reviewed traceability matrices.
* Functional gaps in test plans are reduced to minimum due to completeness of traceability matrices.
* Review of test plan with business users, development team and project’s other stakeholders

1. **Test Strategy**

**Challenges:**

* To monitor test bed /exact hardware-software configurations for the test environment.
* Best use of limited resources/time to provide optimum test coverage to product/project functionality.

**Best Practice:**

* Preparation of a consolidated functional test coverage matrix for various test cases to be executed.
* Utilize an extensive decision model while deciding on the test matrix or each testing cycle.
* Prioritization of test cases in the test plans reduce the risk factors and guide for focusing in-depth on high risk features.
* Review of test strategy with development team.

1. **Test Execution**

**Challenges:**

* Testers struggle a lot while learning the product functionality by using the actual product build for the first time and due to the mismatch between the test plans and the actual functionality
* A large amount of time going towards setting up the test environment due to product complexity
* Issues related to test environment setup and test execution, which take enormous amount of time in resolving for those testers who are new to the test team
* Blocking issues found in the product functional areas lead to extensive re-planning putting the overall testing plan at a stake.

**Best Practice:**

* Set up a process to get the intermediate product builds well before the formal QA handover.
* Establish a knowledge repository of various problems encountered during test environment setup/test execution and their possible solutions.
* Be agile and flexible in execution. Readiness to change plan if projects changing needs and risks warrants.
* Work with development and other teams to ensure install and configuration instructions are detailed enough.
* Revise estimation and Test Plan based on delays on development build

1. **Defect Management**

**Challenges:**

* Incomplete defect description & steps to reproduce which results in too many defects getting re-assigned back for “Needs More Information” from development team.
* Inconsistency in the structure of defects reported by various test team members.
* Testers assigning improper Severity/Priority to defects.
* Too many defects being marked as “Not-A-Defect” by development team due to improper test case execution or not understanding the requirements.
* Inadequate tracking of the various defects.
* Duplicates defects reported by various testers in the team

**Best Practice:**

* Use standard templates for defect entering or defect management tool
* Establish a clear-cut criteria for defining Defect Severity and Priority and train testers on the same
* Enter all relevant information like Steps to reproduce, screen-prints while entering defects
* Triage defects with development and business analyst and determine the next course of action
* Maintain an updated list of the various defects logged against the product till date containing appropriate details
* QA should focus on entering high quality functional defects rather than focusing on entering large quantity of cosmetic defects
* Making high impacting defects visible and escalated so that development team work on right and critical priority defects.
* Collaboration with development and business analyst team

1. **Test Results Reporting**

**Challenges:**

* The test results report should be easy to comprehend and should be useful for all the stakeholders.
* Test results should provide a status of the feature-wise health of the product along with feature-wise defects.

**Best Practice:**

* A comprehensive test results report template should be prepared in a spreadsheet format or auto generated metrics from defect management tools like JIRA, ALM, etc.
* Result Summary report sheet provides the overall summary of the test cases executed and effort spent during the testing for each stage and feature.
* Report should provide the details for the new defects found during the QA and UAT cycle.
* It provides the risk associated with each feature on a scale of High (red), Medium (yellow) and low (green) based on the test cases which have failed, are blocked or were not executed for the feature.
* Defect Severity/ Priority Distribution: This chart provides the distribution of defects based on Severity and Priority.

1. **Test Metrics Collection**

**Challenges:**

* How to measuring the program/product health objectively?
* How to create a mechanism for measuring the efficiency, effectiveness and quality of the testing process so as to identify the areas of improvement?

**Best Practices:**

* Identify a set of process/product metrics to be tracked on continuous basis.
* Develop and share dashboards for these metrics
* Various metrics to be collected are
* Number of S1 and S2 bugs.
* Time to resolve defects.
* Number of defects from Customer (meaning missed by QA during their testing)
* Re-opened defects.
* Regression defects