1.0 Introduction

In today’s fast-paced lifestyle with programmers churning out code in order to make impending deadlines, it is imperative that management receives the appropriate information to make project decisions as early as possible. The Quality Assurance (QA) team is responsible for arming management with detailed information about the quality of the selected solution so they can make these important decisions properly. Management also needs to be equipped with metrics from all aspects of QA and should receive this information as soon as requirements are complete and throughout the project’s lifecycle.

2.0 Metric Categories

Metrics for functional testing fall within three categories: requirements, test case, and defect metrics. In order to determine the true quality of a system, a manager must look at metrics from all three categories. For example, if a manager only looks at the number of defects without determining the number of executed test cases, he or she cannot assess the potential for how many more defects may occur. On the surface, a project with ten defects might sound like it is progressing smoothly; however, if the project has ten defects and the QA team has only executed ten out of two hundred test cases, then the project does not sound as a good.

3.0 How Metrics Fit Within the SDLC

Consider the Software Development Lifecycle (SDLC) depicted in the diagram below. When should the QA team determine which metrics to collect and what threshold to tolerate? When should the collection and reporting of metrics begin? In the past, QA has started after development is complete; however, beginning QA earlier in the SDLC has many benefits. In fact, research has shown that defects found later in the SDLC cost more to correct than defects found earlier in the lifecycle.

Ideally, QA planning begins in the Scoping Phase by determining the types of testing to include in the Implementation Phase. Identification of specific QA requirements should happen in the Requirements Phase, as well as definition of the metrics and tolerance levels for each metric. At the beginning of the Test Planning activities, finalizing the metrics and tolerance levels will occur. Agreeing on the metrics to collect and their tolerance levels ahead of time allows the team to work toward a clear goal, thereby eliminating stress.

The QA team needs to start collecting and reporting the requirements metrics in the Test Planning activities so that they can evaluate the test case writing progression. Collecting and reporting the test case and defects metrics begins in the Test Cycles. All metric
reporting should continue throughout the lifecycle of the project. Having the metrics reported throughout the Implementation and Deployment Phases allows the team to assess the quality and progress of the project accurately. If the team waits until System Acceptance to look at the metrics, it may be too late to make corrections and deploy the solution on time.

4.0 Five Essential QA Metrics

4.1 Requirements Metrics

The most essential of the requirements metrics is the Percent of Requirements Traced to Test Cases metric, which indicates the percentage of tests that have at least one associated requirement. This metric determines the amount of test coverage for the requirements.

\[
\text{Percent of Requirements Traced to Test Cases} = \frac{\text{Total Number of Requirements with at Least One Associated Test Case}}{\text{Total Number of Valid Requirements}}
\]

As the Test Planning activities progress, the percent of requirements traced to test cases will increase. At the end of the Test Planning activities, each requirement should have at least one test case and the percent of requirements traced to test cases should equal 100%. Some requirements may necessitate more than one test case to verify the requirement; however, as a rule, each requirement must have at least one test case. Any additional test cases will not change the percentage. If the percentage of requirements traced to test cases is less than 100% at the end of the Test Planning step, then a review of the requirements without coverage will take place to determine if the requirements are valid and, if so, which essential test cases will aid in verifying the requirements.

Periodically, the project team reviews the Percent of Requirements Traced to Test Cases metric, ensuring 100% continuous coverage throughout the project lifecycle. If an addition, change, or removal of requirements occurs, updating the test cases will take place in order to maintain 100% coverage. If the QA team cannot maintain 100% coverage, a review of the project requirements and test cases will take place to determine why. Incomplete and unclear requirements and resource constraints are common reasons for the inability to maintain 100% coverage. Once the project team defines the reason for the lower percentage rate, the team will take corrective action and incorporate the lessons learned into future projects.

4.2 Test Case Metrics

4.2.1 Percent of Test Case Execution

The Percent of Test Case Execution metric indicates the progress of testing by giving the percentage of test cases executed with the result of a pass, fail, or blocked result. This metric groups tests by test type and/or functional area.
Percent of Requirements Traced to Test Cases = Total Number of Requirements with at Least One Associated Test Case / Total Number of Valid Requirements
Percent of Test Case Execution = (Number of Passed Tests + Number of Failed Tests + Number of Blocked Tests) / Number of Test Cases

The percent of test case execution increases as testing progresses for each set of test cases during execution so that, at system acceptance, it should be 100%. If the percent of test case execution is not equal to 100%, the team must review each unexecuted test to determine the cause of the discrepancy. A review of the defects, test cases, and test results will determine the impact to the project and schedule.

4.2.2. Test Case Pass Rate
The Test Case Pass Rate metric indicates the status of test execution and the quality of the solution based on the percentage of passed test cases. This metric indicates how many tests have passed, grouped by test type, phase tested, and/or functional area.

Test Case Pass Rate = (Number of Passed Tests) / Number of Valid Tests by Grouping

The test case pass rate should increase as testing progresses. If the test case pass rate does not increase, yet stays steady, then it may indicate one of the following: testing is taking longer than expected, blocking defects are preventing testing from progressing, or testing is revealing a significant number of defects. A steady pass rate could also indicate that defects are not being resolved correctly and, as a result, the defect owners are reopening them. If the pass rate does not increase as the project moves forward, it indicates that a review of the requirements, test cases, and defects is essential in determining the need for corrective action to increase the pass rate.

Ideally, the test case pass rate should be 100% at system acceptance. In certain cases, a project team may decide to deploy the solution with a pass rate of less than 100%. The team may decide to defer defects to a future release date or decide not to correct certain defects because the cost of fixing them is greater than their overall impact. The failed tests could also be associated with functionality the team has determined not to be high-risk and, because of that, they were not tested or passed due to their low risk factor. If there are a high number of deferred defects, it may indicate two things: 1.) the functionality was complex and the development team could not implement it during the timeframe given for the project, or 2.) the requirements were not clear enough for proper implementation. Reviewing the project, tests, and defects will help reveal the problem areas that caused the pass rate to
be less than 100% and allow for process changes for future projects.

The QA team should not consider the Test Case Pass Rate metric in isolation; instead, they should review it in conjunction with the Percent of Test Case Execution metric. The percent of test case execution should be at 100% when each test case scheduled for testing is complete. For example, consider the following metrics: the Test Case Pass Rate = 100% and Percent of Test Case Execution = 100%. These metrics would indicate that the quality of the system based on these test cases is good, because each planned test case passed; however, a Test Case Pass Rate of 50% and Percent of Test Case Execution of 50% indicate that testing only occurred for half of the test cases, and of those tests, half did not pass.

4.3. Defect Metrics
4.3.1. Trend Analysis
The Trend Analysis metric indicates the number of open or unresolved defects by date and aids in determining the quality of testing as well as the progress of resolving defects.

There are several trends to watch for in the Trend Analysis Chart. If the line flattens out, it would indicate that testing has stopped uncovering defects. If the line continues to increase after the delivery of all functionality, it would indicate that the QA team is still uncovering defects. If the line does not increase as the delivery of new functionality occurs, it would indicate that the QA team is not executing the proper test cases. Any of these trends warrants a closer look at the testing performed by the testers and the outstanding defects. If the number of defects does not level off but continues to increase, it could mean that the quality is low. For example, consider the following graphs:

![Trend Analysis Chart](image-url)
Trend Analysis 1 shows better progress, because the numbers of defects are leveling off. In Trend Analysis 2, the number of defects is still increasing, which indicates that the QA team is slow to uncover defects or the development team is introducing new defects as they close others. In Trend Analysis 3, the line does not increase very much, which indicates that the QA team may not be focusing on the right areas. In this case, an essential review of the test cases will occur.

The graph indicates that the testers revealed most of the defects during the first two weeks of testing. The testers closed off the defects a few weeks later; however, some defects took longer to resolve. This graph would also indicate that a further review of defects is necessary to determine why certain defects took longer to resolve. The review should include the number of defects reopened, how long it took development to fix the defects, how long it took the QA team to retest the defects, and an evaluation of the clarity of the requirements and design. Based on the review, the team could then make changes to prevent the issues from occurring on a future project.

4.3.2. Defects by Severity
The Defects by Severity metric indicates the number of defects grouped by severity level. The team can utilize this report to get an idea of the number of defects in each level, which would indicate the quality of the solution. Only critical or high-risk defects will affect any
project regardless of project size or risk; however, the QA team cannot overlook the medium- and low-risk defects. Low-risk defects could include defects for typographical errors, which may have a significant impact on some external applications.

| Total Number of Defects and/or Outstanding Defects Grouped by Severity |

At system acceptance, the team will recommend that all critical or high-risk defects be resolved and that only a small percentage of medium- or low-risk defects remain. A large number of critical or high-risk defects would warrant a closer look at the defects, tests, design, and requirements for the project. A review of all open defects should take place before deployment of the system into production.

5.0 Conclusion

When it comes to assessing the quality of a solution, QA Managers may find that they do not have enough metrics, have too many metrics, or they do not receive the metrics early enough in the process. QA metrics are crucial for management to make project decisions based on the quality of the solution at a given time in the project lifecycle. Giving management metrics from each of the three QA metric categories and reporting those metrics in a timely manner will prove to be invaluable for making project decisions.