HOW LONG WILL IT TAKE?

What do you do when you have been asked for a quick estimate for testing an application but you haven't yet had the time to prepare a full test plan?

This can be a daunting prospect when you have little or no control over the quality of the product delivered for testing. It becomes even more difficult if the requirements are poorly defined. Unfortunately, this is a common situation for testers.

Try following these six steps:

1. Develop a basic test structure
2. Apply a "criticality" factor to the tests
3. Apply a "complexity" factor to the tests
4. Plan the number of test runs
5. Estimate durations for the tests
6. Ensure the estimates fall within the project's schedule and resource constraints.

STEP 1. DEVELOP A TEST STRUCTURE

The first step in the estimating process is to define a draft test structure based around system functionality. That is, make a list, in general terms, of what tests or types of tests should be done.

The functionality is commonly defined in a requirements specification. Be sure to supplement the functionality tests with other test types such as interface tests, compatibility tests, performance tests, reliability tests, etc, which may not be addressed in the specification.

In cases where the requirements are not well defined, the use of a test knowledgebase is very helpful. A good testing knowledgebase identifies the technical issues and common patterns of behaviour and failure associated with the implementation of systems of the type under development.

STEP 2. APPLY CRITICALITY FACTORS

Once you have a general idea of what tests you need to do, the next step is to understand the relative importance of each test.

Within any system, some functions will be more critical than others. These include the mission-critical functions (whose failure would negate the purpose of the application), and may include safety critical functions and financial data integrity issues. These functions have a greater impact of failure, and so require testing to a greater depth than, say, minor aesthetic functions (ie, more error cases and combinations).

By applying a simple Critical or Non-Critical factor against each test, the relative importance of the tests and applicable depth of test can be accounted for in the estimate. It would not be uncommon for a Critical function to take more than twice the time to document and execute than a Non-critical function, especially for safety-critical tests.

STEP 3. APPLY COMPLEXITY FACTORS

Another factor in determining test duration is how difficult the functionality will be to test. Not all tests will involve the same amount of effort - some will be complicated to write, set up and run, and others will be straightforward.

Therefore, the next estimating step is to classify each test as being Complex, Medium or Simple. Typically, Complex tests include performance and non-functional requirements, and Medium tests include easy functions that either take a long time to conduct (such as timeout functions), or tests that have a complicated set-up (ie, lots of test equipment or data setup).

A general rule of thumb is that a Medium test takes on average twice as long to write and run as a Simple test, and a Complex test takes twice as long as a Medium test.

STEP 4. PLAN THE NUMBER OF TEST RUNS

Deciding how many times you will need to run each test is difficult if you have no idea on the quality of the software being delivered. Typically, low quality software will require more iterations through the defect finding/ defect checking loop. As a minimum, you should plan to do at least two runs, with the first run to identify defects, and the second run as a regression test following the correction of discrepancies. If you are involved on the project early enough, you should also plan to do one or more dry runs on early drops of software. By limiting the estimate to a defined number of runs, an assumption is applied as to the quality of the product, and this assumption should be highlighted in the estimate.

The level of efficiency gain in multiple test runs is dependent on many factors, including test automation and consistency of testers, with gains of 25%-50% not uncommon for manual tests. Therefore, if your first test run should take 10 days, then you could expect that the second test run will take between 5 and 7.5 days. Test automation has the potential to significantly increase these efficiencies.

STEP 5. ESTIMATE DURATIONS

At this point, simply calculate a duration for each test, accounting for the complexity and criticality factors assigned to each test, with these factors applying equally to all test tasks (eg planning for a Critical test can take twice as long as planning for a Non-critical test).

As a starting point, take a typical Simple, Non-critical test, as these are easier to work with. Think through how long you believe it will take to plan, write, and dry run this test, then add on how long it will take to execute the subsequent planned test runs and to write a report. Then do this for a typical Medium and a typical Complex test, and apply the times to all tests of the same category.

If you are having difficulty thinking through a Medium or Complex test, then apply the “rule of thumb” scaling factor described above to the Simple test to obtain estimates for the Medium and Complex tests.

Finally, apply the criticality scaling factors to arrive at your estimates.

STEP 6. CHECK AGAINST CONSTRAINTS

Having worked out your test estimate, you then need to ensure that the estimate falls within the project’s time, resource and budget constraints.

If it falls outside of these constraints, then some adjustment to the test coverage of the Non-critical functions will be necessary. This will increase the risk of a poor quality product being released to the customer, but you may not have any choice. In the first instance, look at the Complex, Non-critical tests for a reduction in scope. If any adjustment is made (or tests excluded) then ensure that the increased risk exposure to these functions is highlighted in the estimate.

