



The Expert Test Manager: Verifying Third-party Quality

Rex Black, President, RBCS, Inc.

Suppose you went to a restaurant for dinner, sat down, and told the waiter, “Bring me dinner and a drink.” You didn’t provide any further details, though you had something specific in mind. What are the chances that you’ll get the dinner and drink you expected? While no one would ever do this in a restaurant, it happens sometimes on projects that involve third parties.

If we have certain expectations and requirements for an engagement with a third party, those should be defined and clearly communicated between the parties. The best practice is to have that definition and communication before the project starts and to put the agreed-upon terms into the contract. If the third party is delivering software, then these requirements should include quality targets, including measurements of those targets. The measurements should be objective and not subject to distortions.

In addition to defining the requirements, the point at which those requirements must be met should be defined. This can be done by defining entry and exit criteria that establish quality gates for deliverables. Because these quality gates will control the start and end of project phases, they should be synchronized with the phases of the project and aligned with project schedule milestones.

The ISTQB Expert Test Manager syllabus provides a number of examples of entry and exit criteria for various test levels. I have reformatted those in Table 1 and provided my comments and suggestions on implementation or improvement of each criterion.

Type	Level	Syllabus Criterion	Comment/Suggestion
Exit	Unit	Statement coverage meets or exceeds 85 percent.	I prefer to see a standard of 100 percent statement and decision coverage for all new or changed lines of code. I also recommend that automated unit tests, implemented with a specified tool, be deliverables for each unit of code.

Type	Level	Syllabus Criterion	Comment/Suggestion
Entry	Component integration	Code static analysis complete, no outstanding errors.	I would make static analysis of units an exit criterion for unit testing for all new or changed modules. For entry into component integration testing, I would require two or more communicating units that had exited from unit testing. I would also require an approved integration and integration test plan.
Exit	Component integration	All components of functional areas integrated (interfaces verified to be working correctly).	If you can make this happen, another excellent criterion is to have automated integration tests, built with the same tool as the automated unit tests, be deliverables of integration testing. Together with the automated unit tests, you will have a powerful and maintainable regression risk mitigation tool.
Entry	System	No outstanding blocking defects.	This works, but it does require that earlier levels of testing have some type of defect tracking process. Otherwise, a special sanity or smoke test must be run prior to entering system test, with the results of that smoke test determining whether the product is ready for testing.
Entry	System	All known defects documented.	As with the previous criterion, this requires that sufficient information be collected during the earlier levels of testing. Otherwise, the smoke test can be used to establish the known defects, but that really doesn't address the spirit of this criterion.

Type	Level	Syllabus Criterion	Comment/Suggestion
Exit	System	All performance requirements met.	I would suggest that, in addition to performance requirements (i.e., resource utilization, response time, and throughput), all functional and nonfunctional requirements should be met. If any requirement is not met, then a cross-functional team including product and project management should be allowed to accept the problem as a known limitation.
Entry	System integration	No outstanding high priority or severity defects open.	Assuming this test level is preceded by system test, it's reasonable to assume that defects are tracked. However, if different groups are involved, integrating and making sense of the information can be an issue if you didn't or couldn't address the issues discussed in the previous sections on communication and merging test efforts.
Entry	Acceptance	All planned testing by the test group(s) has been completed and the testing results meet the specified criteria.	Of course, the "specified criteria" mentioned in the syllabus must actually be written, measurable, and relevant for this to work. Also, you should be careful to define what "completed" means in terms of testing. Ideally, completed testing has the connotation that all important coverage items were tested, all of the tests pass (or known failures have been officially accepted as limitations), and there are no known defects (other than these accepted limitations).
Exit	Acceptance	Sign off by the accepting parties	I suggest that the sign-off occur after a management review where the results and completeness of the acceptance test are evaluated, discussed, and approved.

Table 1: Annotated Entry and Exit Criteria

It's important to note that Table 1 provides only a small sample of the criteria you would include. On an actual project, you should have a large and thorough set of criteria, addressing various issues that affect the testing work on the project, the test results, and the quality of the software being tested.

As mentioned in the Foundation syllabus, typical entry criteria for test levels should address issues such as the availability, readiness, completeness, and quality of the test environment; the availability, readiness, completeness, and quality of the test tools, including their installation in the test environment as needed; the availability, readiness, completeness, and quality of test items being delivered for test execution; and the availability, readiness, completeness, and quality of the test data. As mentioned in the Advanced syllabus, entry criteria should also address whether the tests are complete and ready to run; whether the tools are available to support test management, defect tracking, and (if applicable) automated test execution; and whether defined approaches for test results logging and tracking, defect reporting, and test metrics analysis exist and are understood by all the testers.

As mentioned in the Foundation syllabus, typical exit criteria for test levels should address issues such as the level of coverage achieved, in terms of code, functionality, requirements, or risks; predicted numbers of defects remaining, defect density, mean time between failure, or availability; cost of continuing versus ending testing; the residual level of quality risk (in terms of known defects, known failed tests, or gaps in test coverage); and schedule targets.

The stringency and formality of the criteria will vary. The product and application domain influences the criteria; for example, safety-critical systems need tougher criteria than a company's promotional web page. The past experience – good or bad – with the third party influences the criteria; for example, if a vendor provided poorly tested software in the past, the rigor of the entry criteria should be increased. The requirements for the system being built and those in the contract or agreement influence the criteria; for example, if usability is central to the value of the software, usability testing and its results should be in the criteria.

In addition to criteria to measure the status of the product, there should be well-defined, objective, measurable project milestones that allow test managers, project managers, and product managers to track the testing and the project against a schedule. To make a milestone measurable, one important thing is to have clearly defined deliverables, with a linkage to the entry criteria for those deliverables. While project or product managers should track the overall schedule, test managers should track milestones, or at least participate in the tracking of milestones, that relate to testing work and the quality of the system under test.

As important as these entry criteria, exit criteria, and milestones are for a single collocated team on a project, when other parties are involved, they become essential bulwarks against chaos and disorder. Therefore, it's important to spend the time to carefully craft the proper criteria and milestones. Not only do all important issues need to be addressed, but the criteria and milestones must be measurable in a way that all sides will agree is objective and conclusive.

It's frustrating to have criteria and milestones that are contested and relitigated by a third party once problems arise with its deliverables. It's also frustrating to find that a third party is trying to find gaps or ambiguities in criteria. However, those situations can happen. For you to deal with these situations, the criteria and milestones must be complete, measurable, objective, and – here comes the tricky part – enforceable. To be enforceable, the criteria and milestones must be clearly traceable to some clause in the contract, if not actually directly in the contract (which is the better practice). There needs to be a defined process that allows the test manager to work with the third party to resolve a violated criteria or missed milestone and, if resolution proves impossible, a defined process to escalate the problem.

During that period of resolution and, if necessary, escalation, the test manager should have clear direction on how to proceed. If testing is not to start or conclude unless certain entry or exit criteria, respectively, are met, then the test manager must have the authority to effectively stop the project, and the test manager must – absolutely must – be insulated from any negative consequences associated with such an action. Be very careful here; we have seen entire testing groups shipwreck themselves – literally put themselves in such a bad way with their colleagues that the test group ended up being dissolved – by allowing themselves to be dragooned into a process cop role, with rigorous entry and exit criteria that they had to enforce on very unhappy coworkers. I feel the best practice is for the test manager to report the status of the criteria to project and product managers and have them make the decision about whether to enforce or waive the criteria.