

Ten Tough Questions to Ask When Developing a Project Test Plan

by Bryan Campbell,

bryan.campbell@valtech.com

www.bryancampbell.com

One reason so many software development projects fail is a poor testing framework to ensure what is developed will actually work in production. Developing a comprehensive test plan for a software development projects requires a specialized skill set but as a senior executive you need to be able to ask the right questions before signing off. This article is part of a series that provides ten tough questions every top executive should ask when reviewing the testing components of a project plan.

Question 1: How much time is spent testing?

One of the biggest reasons software applications are not properly tested is that the time to test the application is not in the project plan. Not placing enough emphasis on testing is common when project managers are trying to meet aggressive timelines. In addition, testing isn't seen as particularly 'glamorous' and either the time to perform it is underestimated or it is minimized with the belief that tools or skilled resources will minimize the amount of time required. Find out how much of your project will be spent testing the application. While there are a range of estimates for the appropriate amount of testing, most business systems will require 20-30% of the project resource total in testing.

Question 2: What is being tested?

Many project plans will identify and resource the testing of the application (known as functional testing) but ignore testing for other important elements of the application. Having an application that performs as required but can only support one user at a time, or takes two minutes to display a new screen probably won't be viewed as a success. Ask what components of the system are being tested. Look for tests that include performance and load tests for the system. Testing for security is often forgotten and can have devastating consequences. Data integrity tests, user interface tests and failover and recovery tests should all be identified and resourced. Make sure the testing addresses your business cycle too, if you process twice as many orders in the month of December make sure this is tested.

Question 3: What testing metrics will be used?

Testing can provide the best insights into whether a project will actually be able to meet its target date and budget. Even if key development milestones are being met the project might actually be behind schedule if this code is buggy or doesn't meet requirements. There are a number of testing metrics that should be communicated during status reports, total defects logged, open defects, defects by type, average time to correct a defect, defects per line of code etc. These metrics should help management assess project progress and quality.

Question 4: Who is performing the testing?

It is often assumed that testing can be performed by junior staff or by the development team itself which is often an attempt to save money, time or both. While the development team needs to develop unit tests to validate the logic of the code that they write, this is only one component of testing. Likewise, junior staff might be able to follow documented test scripts but using them exclusively will not test the application as thoroughly as required. Use a separate testing group ideally comprised of business users and some technical users (to ensure appropriate system tests such as load and security tests). Ask where the testing team will be located and what their experience with the system or business will be. Remote test teams can help test connectivity issues but you will want to ensure they can access the development team for questions. Your testing team will ultimately be telling you whether the application meets requirements and is ready for production, make sure the team is composed of people you can trust and rely on them to thoroughly test the application.

Question 5: How is the testing team aligned with the development team?

Most developers dislike testing because defects mean re-work, however, proper testing can also make sure that what they have developed is the best reflection of their work. The testing and development teams need to work closely to be successful. Ask questions on how the teams will interact, when status meetings will be held between the teams, and whether ground rules for identifying defects been shared between the development and the testing teams.

Question 6: When will testing be done?

In waterfall methodology based projects, testing is usually completed near the end of the project after development has been completed. The advantage of this approach is that it provides a clear delineation between development and testing and makes defect correction easier to manage. Iterative development projects require a more carefully orchestrated approach to testing. While iterative development provides much higher visibility of the development process through frequent deliverables, it requires a similar iterative testing process to continually test the completed application. Iterative development also offers the opportunity to more thoroughly test the application's most important components (from one iteration to the next). Understand which development approach is being applied to the project. The project plan should show when testing will be conducted and what will be tested. If an iterative development plan is being followed, there should be a testing buffer every three iterations to ensure all defects can be baselined and corrected.

Question 7: How did you determine the correct allocation of effort and time for testing tasks?

Similar to estimating effort for development, testing effort should be empirically determined. There is a direction relationship between development effort and testing effort, this should be reflected in the plan. Three months of development effort with only two weeks of testing will not thoroughly test the application. Some testing might be automated which can reduce the time required to perform a test but not necessarily the effort (creating and maintaining automated test scripts can be resource intensive).

Question 8: How will tests for the system be developed?

The most significant software defects are not caused by poorly written code but rather poorly defined requirements that are not properly tested. There should be traceability between requirements, code and tests. Test cases should be created from use cases or similar documentation. Ask how tests will be developed and look for dependencies between requirements, code and test case development.

Question 9: What automated testing will be used?

Developing automated test scripts can relieve labor intensive and onerous tasks from project members as well as increase regression testing for the application (the process of retesting the application after a change has been introduced). However, developing automated tests require special tools which can add costs to the project. In addition, these tools need support and the scripts themselves must be updated and maintained anytime the code or data changes. Often times, automated test tools can be integrated with requirements management and defect tracking tools. Ask who will be responsible for maintaining the automated test scripts and make sure it doesn't cost you more to support these tools than it would to simply hire the labor to perform the tests.

Question 10: How will defects be tracked and managed?

As your project progresses the number of defects identified and corrected will grow. A good defect management system is needed to capture, manage and report these defects. For larger projects the project plan should identify a test coordinator to assign and track defects. Entering defects takes time so ensure that only the most important information is captured otherwise the costs of using the system will skyrocket and users will be reluctant to use it. The defect tracker

should also support discussions between the test and development teams if there is need for further detail or to reclassify a defect.

About the Author

Bryan Campbell is an Engagement Manager with [Valtech Inc.](#), a software development company specializing in object oriented developing using the Unified Process. Mr. Campbell has spent more than 15 years in the information technology industry across a wide range of disciplines ranging from network operations to enterprise architecture to software development. Currently completing his MBA specializing in Information Technology Management, Mr. Campbell has managed a series of large scale IT projects including software development projects using iterative development techniques. Mr. Campbell has worked in a variety of industries including Telecommunications, Utilities and Insurance. He is the author of several articles on software development and has presented at several conferences. More information and articles can be found at www.bryancampbell.com.