

Senior test architect David Johnson describes the distinction between functional and performance requirements and the benefits from keeping these two test efforts separate. Examining differences in test investment, test ROI and risk, Johnson makes clear the importance of understanding the distinctions and testing accordingly.

Many organizations implement requirements based testing for both business functionality (functional requirements) and performance criteria (performance requirements). The challenge becomes keeping a clear distinction between these two discrete testing targets – functional vs. performance. Functional testing and performance testing can harvest significant returns for the organization but when the two are "mixed" the investment cost will increase while the return on investment (ROI) will often decrease - combining the two can also lead to unnecessary risks being inserted into the project lifecycle.

In this article, we will provide a definition for requirements, functional requirements and performance requirements. We will provide examples of both functional and performance requirements and discuss the differences between them. We will then discuss why it is important to distinguish between them from a testing investment, testing ROI and risk perspective.

What is a requirement?

A requirement is a capability or function that must be delivered by a system component or components.

What is a functional requirement?

A functional requirement is a specific business need or behavior as seen by an external user of the system.

Basic examples:

- Logon shall require a valid User Id, User Password and User Domain.
- Three invalid logon attempts shall result in the current session being locked out for five minutes.
- Six invalid logon attempts by a unique User Id shall result in the User Id being disabled.

What is a performance requirement?

A performance requirement specifies the speed or operational effectiveness of a capability that must be delivered by the system architecture as seen by the technical layers within that system architecture.

Basic examples:

- Valid logon transaction response shall occur within 999 milliseconds of the request when the system architecture is under nominal and peak transaction loads as itemized by the transaction profile.
- Invalid logon transaction response shall occur within 999 milliseconds of the request when the system architecture is under nominal and peak transaction loads as itemized by the transaction profile.

Functional vs. performance: Requirements

Functional requirements address the needs and behaviors required by the user community while performance requirements address the speed and effectiveness of the overall architecture. While most testing organizations are accustomed to dealing with functional (business) requirements many do not have the same level of expertise when dealing with performance requirements. The critical difference between these two types of requirements and testing them is that functional requirements deal with the business while performance requirements deal with the architecture. The following table illustrates some of the differences between functional and performance testing – it is important to realize that requirements will reflect these fundamental differences.

	Functional Testing	Performance Testing
Testing Focus is on...	Functional Business Events	Individual Transactions
Test Environment...	Must support Business Functions	Must mimic Production Architecture
Test Cases...	Describe detailed business events	Describe transaction flow

Test Success based on...	Simple pass/fail functional criteria	Transaction Response times & Capacity Consumption <ul style="list-style-type: none"> • CPU • Memory • .NET thresholds • Etc.
Test Cases executed...	Separately or as a package	Almost always as a packaged set of tests/transactions (Scenario)
Test Cases designed to...	Execute once per test cycle	Execute several hundred if not thousands of times per test cycle
Execution is a must when...	The Business events change	The architecture changes

Functional vs. performance: Investment & ROI

Earlier we stated that mixing functional and performance requirements testing would increase the overall investment while decreasing the return on the investment. Why would that be the case? It would seem that by combining these two aspects of the testing space a testing organization would become more efficient thus increasing the ROI. We will look at the investment and the resulting ROI from a people, process and technology perspective, noting the differences between functional and performance requirements and the impacts on testing those requirements.

People & process

The skill sets required to support the creation and subsequent testing of functional requirements is very different than those required for performance requirements; therefore, it is much easier to manage and track the resources and deliverables separately. Separation also leads to efficiencies, especially from the perspective of performance. Identifying and then testing performance success criteria requires a substantial "people" investment on the part of the overall IT organization – one that is often not required for every functional/business change that occurs. Keeping functional and performance requirements and supporting testing artifacts distinct provides the flexibility of executing

the appropriate level of testing given the current level of risk. For example, if changes in the business functionality have not had a significant impact on the transaction mix or behavior, then performance testing may not be required. On the other hand, if the architecture has changed (i.e. introduction of VMware) but not the business functionality, then a "light" functional test followed by intense performance testing would be appropriate.

Technology

The technology "tool-kit" required to support the testing of performance testing is much more extensive than what is required to execute functional testing. The functional testing environment must support a relatively small volume of business events – the focus is on testing these events from birth-to-grave. The performance testing environment requires production size (or greater) volumes of business transactions being executed in a production like environment – the focus is on testing all aspects of the architecture. It is important that the profile of the test environment matches production, not just its capacity. For example, if production applications run on standalone servers it would be inappropriate to execute performance test on a VMware based platform – not because VMware is more or less efficient but simply because VMware is an entirely different architecture.

The functional testing "tool-kit" includes test cases and the ability to execute and track the results of executing these test cases – this can range from a simple spreadsheet to scalable enterprise testing frameworks that include requirements, test cases, test execution, defect tracking, and test reporting. The performance testing "tool-kit" includes test scenarios and the ability to execute and track the results of executing these scenarios – this is a much more extensive exercise than simply tracking a pass/fail business event, the behavior of the entire architecture must be tracked (transaction times, network load, network latency, CPU consumption, .NET counters, database response, etc.). In order to provide a controlled load, a performance tool that can mimic the transactions of thousands, if not hundreds of thousands, of users will have to be used.

Functional vs. performance: Risk

We have seen some the differences between functional and performance requirements and the testing of those requirements. The people, process and technology required to address these requirements have a different focus – business events (functional) and architecture (performance). Why would combining the two increase operational risk? There are two basic reasons why this combination would increase operational risk. First, the combination will lead to unnecessary

complexity in terms of test planning and execution which will in turn increase opportunities for missing key testing objectives. Second, the objectives are very different – functional testing should test all aspects of the business which requires both breadth and depth. Performance testing should test all aspects of the architecture which requires breadth but may not (often does not) require depth. Creating too much depth/complexity within the performance testing space may introduce unnecessary complexity that could hide or mask architectural issues.

Test requirement myths

Functional and performance requirements are the same

-Functional Requirements address business events and business functionality.

-Performance Requirements address architectural speed or operational effectiveness.

Functional and performance testing can be done "together"

-While they certainly can be done "together" the testing objectives and supporting

People, Process, and Technologies are altogether distinct.

Functional and performance testing can be accomplished by the same resources

-The skills and experience required for functional testing are distinct from those required for performance testing. These are separate roles, a resource may be capable of performing both but this does not mean they should be treated as one activity.

About the author

David W Johnson (DJ) is senior test architect with over 22 years of experience in information technology across several industries. He's played key roles in business needs analysis, software design, software development, testing, training, implementation, organizational assessments and support of business solutions. Johnson has also developed specific expertise over the past 12 years on implementing "test ware," including test strategies, test planning, test automation -- functional and performance -- and test management solutions.

This was first published in [October 2010](#)