Chapter 9
Software Testing in Systems of Large Scale

Wojciech Filipowski and Marcin Caban

Abstract. Testing plays an important role in the process of application a change to an IT system, no matter if the system or the change are simple or sophisticated. The articles covers general topics on: the role of testing, classes of tests and test planning. It also gives an overview on how different test categories can be fitted into the implementation process and what the typical classes of tests are, that play most important role in the process.

9.1 The Role of Testing

Testing plays quite an important role in the process of application a change to an IT system, no matter if the system is complex or not and the change is simple or sophisticated. The success factors, however, are:

- to chose a proper type of test/tests to the unique situation taking into account: all the possible aspects of the change, the system and the specificity of an enterprise - the client;
- to perform the chosen tests as planned with the agreed level of services when it comes to delivery of bug fixes.

There is one more thing so obvious, that is sometimes forgotten or not taken into consideration seriously enough in the whole rush and tension of the project, that needs to be mentioned here to put more emphasize on planning role and preparation for the testing in the implementation process. The thing can be just turned into one simple question: "What is testing for"? And the answer is on one hand - obvious, and on the other hand - difficult to give, since when one tries to explain it, there are more and more arguments coming in, making it hard to drain. Nevertheless, in this article...
the authors want to summarize the most important reasons to: understand the great need to perform tests in a project, and the necessity to plan them choosing a proper set of tests each time we run a new change in the system (project of implementation).

Now, let us try to give the answer to our question. To describe it, let’s define the area of analysis: since faults or errors while developing and implementing a change in an IT system can emerge in every part of the system, no matter if it is the hardware or software. To simplify the deliberation without any harm to the merit, we won’t go too deep in the technology, but rather describe the topic from the project management point of view, enabling the authors derive from their experience in the field of project management. First part of the answer is frankly as simple as the question itself: "Testing is the mean to search for, find and correct faults" [1]. This is correct of course, but it is only a part of the complete answer, so we want now to elaborate a bit on it. As we agreed, testing shows bugs. Now: what are bugs, why we want to find them and what kind of bugs are we able to find? Bugs are any undesirable behavior of the system, for instance: faults in functionality, system performing contrary to the requirements or its purpose, poor efficiency, errors in data operations, faults in configuration, communication or integration errors, and so forth. All this: makes the project plan being more difficult to control, causes the project to be less cost effective and more resource consuming, aggravates system security [2] and at the end puts a risk for business. That is why it is so important to find and fix most bugs possible before the system runs in the new version at the client’s area of business. Finally, there are different bugs to find, in nearly every area of the system, so to manage the "search for faults" in an orderly manner, we need to plan to perform several sorts of tests, which are introduced below.

9.2 Types or Classes of Tests

There are numbers of types of tests that are performed on different stages of a project. In this section primary classes of tests will be characterized.

**Unit Test.** Chronologically, during the implementation phase, when engineers change the source code, the Unit Tests are performed. These tests aim at most simple bugs at the lowest level of implementation work e.g. syntax source code bugs, or logical misunderstanding of requirements. Unit Tests scenarios mainly operate within a single library in terms of source code or within limited area of developed functionality to enable most detailed bug search and fix [3].

**System Test (SYS).** More complex are the System Tests. These operate from a little wider perspective, i.e. system as a whole - perspective. In addition or as a follow-up to the Unit Tests, System Tests explore the whole functionality delivered within the change and that is why they can find bugs in the links between code libraries, layers of the system or in the area of system ergonomics.

**Integration Test (INT).** While the system test operates in the area of a single system only, in large integrated environments, to check the reliability of the whole