



## Summary

# Title: Methodology for testing nonfunctional requirements of server applications

**Author: Petya Vasileva Mitovska**

**Scientific supervisor: Sylvia Ilieva, Ph.D.**

**Advisor: Iliana Manova**

**Date: 2007, 17th October**

*Keywords: Testing, types of testing, nonfunctional testing, SDLC, testing automation, GQM, load testing, results analysis*

Testing has always been an important part of the software development. The evolving of different Software Development Life Cycles (SDLC), urges the improvement of all methodologies used in quality management. The main goal of the techniques used during the testing phase is to assure that the whole functionality of the product is fully covered and the time spent is as short as possible. The role of testing is not only validating and verifying system behaviour but also uncovering problems even before they have been implemented. Along with the responsibility to check the software upon functional requirements, testers also have to keep track of nonfunctional characteristics like usability, response time, security, reliability. Due to the critical effect of time upon the success of a certain product, the test team often chooses to automate part of the testing process. Automation is preferred for its ability to shorten time and to reduce the number of resources needed. It also has greater accuracy of results and guarantees that precisely the same actions are done in each test.

This master thesis presents a methodology for planning, execution and evaluation of the results from testing nonfunctional requirements for server applications. In the first chapter is discussed the role of quality assurance and its place in the SDLC. Different types of testing have been classified according to:

- The phase of software development
- The functional and technical requirements for the system

The methodology offered in this master thesis is applied to a server application, which had been described in Chapter 2. All the software techniques used in the project are reviewed with their advantages and disadvantages. The functional scope of testing is defined.

A goal-oriented metric (GQM), is used to prepare a plan for the testing of the nonfunctional requirements described in the previous chapter. This metric consists of 10 steps that transform business goals into software measures:

1. Identify Your Business Goals
2. Identify What You Want to Know or Learn

3. Identify Your Subgoals
4. Identify the Entities and Attributes
5. Formalize Your Measurement Goals
6. Identify Quantifiable Questions and Indicators
7. Identify the Data Elements
8. Define Your Measures
9. Identify the Actions Needed to Implement Your Measures
10. Prepare a Plan

Each of these steps is applied to the server application project in order to demonstrate the process of preparing a test plan with particular business goals.

The actions of virtual users in each load script are described after a short introduction to the abilities of the tool used for implementation of the tests – Mercury LoadRunner 8.0. Using the acquired results, the main parameters are presented as functions of the number of simultaneous virtual users. As a sequence of the analysis made in Chapter 4 the following future activities for the process of testing should be considered:

- Expand the values used in the measurement process in order to reach the maximum of system performance
- Examine the system performance when the software configuration is changed
- Examine the system performance in different hardware environment