CENG 492

Test Specifications Report

CennetElması

Mustafa İlhan

M. Elgin Akpınar

Osman Tahsin Berktaş

4/23/2011

Table of Contents

1. Introduction

This document describes both the test plan and the test procedure of the product SNMP Agent and Network Simulator, developed by CennetElması. After members develop a new feature, this feature should be tested until it's correctness is ensured.

1.1. Goals and objectives

The purpose of this document is to give information about the scope, resources, strategies and procedures of testing management as preparation for final release of SNMP Agent and Network Simulator. By extensive testing, it is intended to have a product free from defect and guarantee that project modules perform as expected. This document is also going to be a guideline for software reliability, quality and maintainability issues of this project.

1.2. Statement of scope

This document will explain the measures CennetElması is taking to make SNMP Agent and Network Simulator a well-functioning, robust and well-maintained product. To achieve this, we will elaborate on

- what we will test,
- what are our constraints while testing,
- how will we handle severe, user-facing bugs,
- how will we find internal bugs,
- how our testing relates to our schedule,
- how team members are responsible for different tests.

1.3. Major constraints

1.3.1. Time

There is nearly one and a half month to complete the project. The remaining time interval is mostly occupied by implementation of the project. Hence, implementation and testing progresses should continue in parallel. Whenever a module is completed, or some modules are integrated, required unit tests and integration tests will be done immediately.

1.3.2. Usefulness of Obtained Test Results / Cost Ratio

The ratio of usefulness of test results/time is the most important factor to determine whether a test is worth the time. Giving too much time to test some modules will be necessary when we consider the “Test results/costs” since working on testing process requires both time and labor. In order to be efficient we will first do the most important tests, than the others.

1.4. Definitions, Acronyms and Abbreviations

SNMP: Simple Network Management Protocol
IDE: Integrated Development Environment
MIB: Management Information Base
GUI: Graphical User Interface

1.5. References

METU Ceng492 Spring 2010 Software Testing Specifications Template
IEEE Standard for Software Test Documentation

2.Test Plan

First of all, in this section, whole testing plan of SNMP Agent and Network Simulator is described. In other words, past, current and future actions about testing procudures are mentioned in this document. Each member of CennetElması is responsible from his own distributed work. To have a stable development process, each member actively uses Google Code Issue Tracking system in case of any work to be done. Thus, testing works are also distributed to each related member.

2.1. Software to be tested

SNMP Agent and Network Simulator has four components which are HTML Renderer, Persistence, SNMP, Simulation Engine. All of these components will be tested thoroughly.

2.2. Testing strategy

2.2.1. Unit testing

HTML Renderer, Persistence, SNMP, Simulation Engine components will undergo a unit testing procedure. The purpose of this phase is to ensure that each component does not have any internal errors. A black-box testing strategy will be applied during the testing process.

2.2.2. Integration testing

Firstly SNMP component will be integrated on Simulation Engine and then SNMP will added to them. After all these integrations our server side will ready to integrate to client side which is generated by HTML Renderer. In other words lastly server side and client side will be integrated.

2.2.3. Validation testing

According to IEEE Standard Glossary of Software Engineering Terminology, Software Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

Our Software Requirements Specification Document defines the functional and non-functional requirements of the SNMP Agent and Network Simulator Project. Every single requirement in this document must be considered one-by-one and the software must be guaranteed to satisfy all these requirements. Test cases and scenarios are defined and will be run to ensure the correct working of the project as a whole. To test the conformity of the non-functional requirements, performance tests will be run using profiling tools.

The order of validation is the same as integration testing.

2.2.4. High-order testing

2.2.4.1. Performance Tests

This product will be tested with different number of devices. In this test case, different simulation scenarios' load on hardware in manner of consumed CPU power and memory will be tested. This test is related to the Simulation Engine, SNMP and Persistence component.

2.2.4.2. Stress Tests

In stress testing devices will be tested to send large number of traps and receive that sent traps. This test is related to the Simulation Engine and SNMP component.

2.3. Test metrics

In this testing process, following metrics will be used:

- Number Of Test Cases Executed
- Number of Test Cases Successfully Executed
- Number Of Bugs Detected
- Number Of Bugs Fixed
- Number Of Priority Bugs Fixed

2.4. Testing tools and environment

During testing process, following tools and environments will be used.

Ireasoning SNMP Trap Receiver: SNMP Component of this product sends SNMP Traps, so Ireasoning SNMP Trap Receiver will be used to ensure that sending trap process achieved.
Eclipse IDE: The debugger plug-in of Eclipse includes most of its UI, and the ability to delegate to a specific debugger implementation at the user's request.
Google Code: Google Code Issue system for managing test and bug report will be used
Browsers: This software is a web application and users will access to this product via browsers. Hence this project should be tested in Internet Explorer, Opera, Safari, Chrome and Firefox to ensure that it works correctly.

This project is sponsored by Siemens and they will be take part in testing of this software.

3.Test Procedure

In this section; strategies and procedures that are going to be followed in the test process are explained.

3.1. Unit Test Cases

3.1.1. HTML Renderer Component

HTML Renderer Component constructs the GUI of the system. During the testing process, it will be tested whether all GUI components responses as if when are supposed to do, by running on different browsers.

3.1.2. Persistence Component

Persistence component is the database component of the system. Therefore, it will be tested whether this component writes to xml documents and reads from xml in a proper way. In addition to these MIB files will be tested whether it is successfully parsed, alarms extracted and listed correctly.

3.1.3. SNMP Component

SNMP component is the component which formats the Trap coming from simulation engine and sends it to Network Management Tools such as HiPath Fault Management. During the testing procedure, it will be tested whether SNMP component interacts between Network Management Tools and interior components.

3.1.4. Simulation Engine

Simulation Engine, basically, retrieves input from HTML Renderer and Persistence, then runs the simulation. Finally, it sends the results to SNMP component. It will be tested whether simulation engine component gets the inputs from HTML Renderer and Persistence, simulates the network without any fault and lastly sends the results correctly.

3.2. Integration Testing

As discussed in section 3.1, the system consists of 4 major components. One of them, the HTML Renderer component is on client side of the system, while the others are on the server side. Therefore, there exist interactions between components as there exist communication between client and server side.

During test phase, it will be tested whether this communications between components maintain as foreseen in the design steps.

3.3. Validation testing

Validation Testing cases are followed:

1. What happens if the user runs a simulation without any devices?
2. What happens if the user be offline during a simulation?
3. What happens if all the devices send Trap at the same time?

Expected Values for above cases:

1. The system will warn user and simulation will not start.
2. Simulation will continue in the server and when user online again can see the alarms fired after user logged in.
3. All trap sending request will be controlled by a thread and will be send using a queue which is filled by the other threads which represent devices. In addition to these filling queue is controlled by semaphores. Therefore devices will add their requests to the queue and sender thread will send them sequentially.

3.4. High-order testing (a.k.a. System Testing)

Higher order testing consists of two phases, which are performance testing and stress testing.

Since the system has only one user, number of users which use the product at the same time is not a considerable issue. However, the number of devices which are added to the system before the simulation begins, may effect the performance of the software. During performance testing, program will be tested with different amount of devices and checked whether it acts in the proper way.

In order to draw the limits of the product, it will be run with a large number of devices. Moreover, it will be run so many times repeatedly. In reduced time simulation, the reduction will be tested to see how much it can be scaled.

4. Testing Resources and Staffing

In our project, JUnit feature of Eclipse IDE is used. While implementing the project, the fresh code is tested simultaneously. After testing the new prepared feature, if it passes all the testing steps, it will going to be integrated to the whole system. However, apart form JUnit, sponsored company, Siemens E.C will be asked for consultance and end-users who are unfamiliar with our project will be needed. The general features of product will be explained to end-users.

5. Test Work Products

Whenever a tester finds a bug, he will assign the mission of solving the problem to a relevant member, by using SVN application of Google Code. By doing so, it will be very easy to keep track of missions assigned to each member, as far as the bugs which are resolved or not.

6. Test Record Keeping and Test Log

So far, the approaches that will be applied in testing the project was explained and the data of these test must be stored in a log.

The structure for storing test results consists of chronologically stored test cases, what order we executed these, who did these tests and whether it is failed or passed. Test records are going to be stored in code.google.com/p/hurma under issues part. Evaluating test results depends on the part of the project. Client part will be tested by end-users. Other parts will be tested by the members of our sponsored company.

Another testing log is the blog of the project group. After each testing, a blog content will be published, explaining the testing procedure and the results. In case of any failure, possible causes of the problem and the methods to solve it, will be discussed.

7. Organization and Responsibilities

Unit Tests: Osman Tahsin Berktaş, Mehmet Elgin Akpınar

Integrations Tests: Mustafa İlhan, Osman Tahsin Berktaş

Performance Tests: Mehmet Elgin Akpınar, Mustafa İlhan

Feature Tests: Osman Tahsin Berktaş, Mehmet Elgin Akpınar

Validation Tests: Mustafa İlhan, Osman Tahsin Berktaş

System Tests: Mehmet Elgin Akpınar, Mustafa İlhan, Osman Tahsin Berktaş

8. Test Schedule

Test Specication Report:	24 April 2011
Unit Tests:	1 May 2011
Integrations Tests:	5 May 2011
Performance Tests:	10 - 15 May 2011
Feature Tests:	15 - 20 May 2011
Validation Tests:	20 - 25 May 2011
System Tests:	25 - 30 May 2011