TEST SPECIFICATIONS REPORT

by

AjanSimit
# Table of Contents

1. Introduction ................................................................................................................................. 2  
   1.1. Goals and Objectives ............................................................................................................. 2  
   1.2. Statement of Scope .................................................................................................................. 2  
   1.3. Major Constraints .................................................................................................................. 2  
      1.3.1. Time .............................................................................................................................. 2  
      1.3.2. Data .............................................................................................................................. 2  
      1.3.3. Hardware ...................................................................................................................... 3  
   1.4. Definitions, Acronyms and Abbreviations ............................................................................. 3  
   1.5. References ............................................................................................................................ 3  

2. Test Plan ...................................................................................................................................... 4  
   2.1. Software to Be Tested .......................................................................................................... 4  
   2.2. Testing Strategy .................................................................................................................... 4  
      2.2.1. Unit Testing ................................................................................................................... 4  
      2.2.2. Integration Testing ....................................................................................................... 4  
      2.2.3. Validation Testing ......................................................................................................... 4  
      2.2.4. High-order Testing ....................................................................................................... 4  
   2.3. Test Metrics ......................................................................................................................... 5  
   2.4. Testing Tools and Environment ............................................................................................ 5  

3. Test Procedure ............................................................................................................................. 6  
   3.1. Test Unit Cases ...................................................................................................................... 6  
   3.2. Integration Testing .............................................................................................................. 6  
   3.3. Validation Testing ............................................................................................................... 7  
   3.4. High-order Testing ............................................................................................................. 7  

4. Testing Resources and Staffing .................................................................................................... 7  

5. Test Work Products .................................................................................................................... 8  

6. Test Record Keeping and Test Log .............................................................................................. 8  

7. Organization and Responsibilities ............................................................................................... 8  

8. Test Schedule ............................................................................................................................. 8
1. Introduction

This document is the test specifications report for SNMP based Network Simulator for the group AjanSimit.

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 this project. By extensive testing, it is intended to have a product without any bugs and that works smoothly. The project consists of several components and in order for the components to work in a harmony, the components need to be tested first individually and then while integration to verify if they are communicating properly and working as expected. Therefore, testing is very critical. Also, this document is going to be a guideline for software reliability, quality and maintainability issues of this project.

1.2. Statement of Scope

This document covers descriptions of the methods used in the testing process of the network simulator, plan of the testing phase, procedures to be followed and techniques of keeping records during testing. During the development of project, each component is to be tested by our developers just after it is said to be “done”. However, since many components work interactively, further testing of the interaction and function of the components and testing of the whole system is obliged.

1.3. Major Constraints

1.3.1. Time

Team members of this project are subject to intense academic work that is not related with the project and this has a huge effect on the overall development situation of the project. The implementation of the project is still going on and there is time needed for the testing phase. But since there is less than two months left, time is the greatest constraint for testing.

1.3.2. Data

Since the configuration files for the devices of the network to be simulated are given by Siemens EC, the testing phase is strictly dependent on the data given by them.
1.3.3. Hardware

The architecture of the computers used in testing phase should have enough capacity in terms of memory and processor speed to run the simulation many times. A steady internet connection is also necessary to test the simulation.

1.4. Definitions, Acronyms and Abbreviations

- **MIB**  Management Information Base
- **SNMP**  Simple Network Management Protocol
- **Trac**  Issue Tracking System

1.5. References

- AjanSimit, Revised Design Report, 2011
- IEEE Standard Glossary of Software Engineering Terminology
2. Test Plan

2.1. Software to Be Tested

The software to be tested is an SNMP based Network Simulator which is currently in the development phase.

2.2. Testing Strategy

Each member of AjanSimit is responsible for their own distributed work. To have a stable development process, the developers actively use Trac system. So, both the developers can test their own work and the testers can test the developed code.

2.2.1. Unit Testing

All the components of the system are tested both during the phase of development and afterwards. The developer tests the unit while developing the code and after it is complete, the main tester of the team conducts a thorough test on the unit.

2.2.2. Integration Testing

After a component is tested and works properly, the integration phase begins. The components that will communicate directly are integrated and tested to see if they are working and communicating as expected. When the whole system is developed, integrating and testing it will be more time consuming. The integration is done by the developers but the integration testing is done by the tester of the project.

2.2.3. Validation Testing

In the validation testing phase, evaluation of the system or any of its components will be tested during or at the end of the development process to determine whether it satisfies specified requirements. So, the implementation of the components will be tested to see if they fit design reports.

2.2.4. High-order Testing

Several high order tests will be applied by the test team. Performance tests will be done to have an idea of the load the program can handle. Also, it will give an idea of the system’s running speed. Then, stress tests will be applied to test the scenario runtime with possible maximum unit on it and maximum interaction between them. So, the system will be tested for crashes. Lastly, alpha and beta tests will be done when the project is in near
completion to use and test all of its features, improve the product and help in releasing a complete and error-free product.

2.3. Test Metrics

The following metrics will be used as criteria for testing:
   a- Number of Test Cases Executed
   b- Number of Bugs Detected
   c- Number of Bugs Fixed
   d- Number of Priority Bugs Fixed

2.4. Testing Tools and Environment

Testing will be done in NetBeans environment. There are testing tools provided such as “JUnit”, “XTest” and “Jemmy”. The tester team of AjanSimit will mostly be using “JUnit” for both unit and integration tests. For the client web application, “AppPerfect Web Test” will be used in regression tests.
3. Test Procedure

Testing has a great significance in project development. In SNMP simulator development, testing is a live component of development. That is, testing is done whenever a new unit is introduced to the system. There are some testing cases and methods which are explained as follows:

3.1. Test Unit Cases

As a new unit introduced to the system, that unit always needs to be validated independently. Most of the time, we assume the smallest testable unit as a method of a class. So, each method in the source code is tested to check that it behaves as it is designed to behave right after it is implemented. As an example for this approach, when we developed FileUploader (for entering new data to database) first we check whether it can upload the file successfully and whether our parser can parse a VirtualMap file (.xml) or not. Therefore, test activities of unit cases are performed while implementation is underway.

Regression testing is the main approach for unit testing. Each newly written piece of code should be tested to check it works fine with the rest of the code.

3.2. Integration Testing

Integration testing is a testing method which test whether the components can run integrated and together or not. If we keep going with the example above, if the file can be uploaded to the server and the server can parse .xml file then integration test should test a file uploaded to the server which can be parsed directly after it is uploaded instead of needing any interaction to combine these two.

We see “Simulation Engine” component is the core element of whole project. Other component of the project will be integrated to this core component one at time. Securing one component is fully integrated to the Simulation Engine, integration testing of another component to the formerly integrated compound, which consists of Simulated Engine and several integrated component, is initiated until all the component is fully integrated to the system.
This bottom-up-like approach makes us able to start integration testing process after development of one component is completed. This way, we don’t have to wait for integration test until all the components is completed.

3.3. Validation Testing

Validation testing is essential when a task is carried out by a software tester. The aim of this testing is to check, if the software has been made in lines with the requirements of the client.

In validation testing process, prepared inputs which is arranged in the a way that represents possible actions of intended users will be given to the system, then outputs will be examined to see whether they satisfy the requirements of the client.

We’d like to provide a release candidate version of the project to our sponsor company Siemens EC to have their feedback before the final product is offered. This subject will be discussed in future meetings with Siemens EC.

3.4. High-order Testing

System testing has the same mentality with the integration testing but here we are checking the whole system instead of just two or three components.

In context of high-order testing, responsiveness and reliability of the system will be tested. The maximum number of virtual devices that the system can handle, the average response time to user actions, the required time to parse MIB and XML files are some of the subjects which will be tested through this process.

4. Testing Resources and Staffing

The team’s main tester is Egecan Yılmaz. Unit tests will be done by both that unit’s developer and Egecan. Integration tests, validation tests and high-order tests will be done mainly by Egecan and Shafi Gonesh when needed.

We are using free edition of “IReasoning MIB Browser” application to test that SNMP packets sent by source code actually reaches to their destination without any corruption. We do not estimate that any additional resources will be needed through testing process.
5. Test Work Products

The result of the testing process is both the identification and correction of the bugs, and having a stable and working product.

6. Test Record Keeping and Test Log

When the tester finds a bug, he will assign the correction to the developer of the corresponding module using TRAC. So the records of the tests and corrections will be automatically kept and logged.

7. Organization and Responsibilities

Since delivering code free from bugs as much as possible is the developers’ responsibility, unit testing of the source code is done by the person who writes the code. Tester of the team keeps track of these unit tests done by developers. Integration and validation testing is the main responsibility of tester of the team. Other testing activities will be distributed to the team members by the tester of the team.

8. Test Schedule

<table>
<thead>
<tr>
<th>Test Type</th>
<th>Date Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Tests</td>
<td>Until 15.05.2011</td>
</tr>
<tr>
<td>Integration Tests</td>
<td>Until 25.05.2011</td>
</tr>
<tr>
<td>Validation Tests</td>
<td>15.05.2011 - 25.05.2011</td>
</tr>
<tr>
<td>High Order Tests</td>
<td>25.05.2011 – 01.06.2011</td>
</tr>
</tbody>
</table>