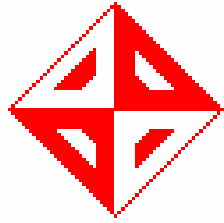


Software Configuration Management Plan

February 26, 2006

SmartGuardian

SmartGuardian™ Software



Computer Engineering Department
Middle East Technical University

by

SmartSoft Network Solutions, Inc.

Tayfun Şen - Özden Meren - Sedat Zelyüt - Hakan Özadam - Akın Öztürk



TABLE OF CONTENTS

1	INTRODUCTION.....	1
1.1	Purpose of the Plan	1
1.2	Scope.....	2
1.3	Definition of Key Terms	3
1.4	References.....	3
2	SCM Management.....	3
2.1	SCM Organization and Responsibilities	3
2.2	Applicable Policies	4
2.3	Development Environment and the Infrastructure.....	4
2.4	The Configuration Items	5
2.5	Configuration Control	7
2.5.1	Change	7
2.5.2	Bugs	8
2.6	Schedule.....	9

1 INTRODUCTION

1.1 Purpose of the Plan

“You can't step in the same river twice.” Heraklitos once said to explain the importance of the change itself. The second time would mean that both you and the river have changed. This was in the 500 B.C. and change is as important as it was in those days, if not more important in these years in which the technology is advancing rapidly. Change is important in the software development too since no software is “bug-free” and the development itself is a process removing bugs and introducing new ones as a result of the coding. Thus the software configuration management plan comes into play. To plan the process of change and to perform it effectively, a formal mechanism must be defined; this is presented in this configuration management plan.

Software Configuration Management Plan is a document of methodology which aims to help project members to identify, manage, control and audit the changes properly during implementation of the software project. Lifecycle of a project consists of changes, updates, tests and releases. In addition, members of the project participate in these activities separately or in groups. That is why, there may exist conflicts, confusions and inconsistencies during development. In order to prevent such situations we have prepared this plan which includes the details of the components of the software, how they are kept, updated, built and released.

The intended audience of this software configuration management plan is the developers of the SmartGuardian project. It will be used in change control throughout the development process.

1.2 Scope

This report documents the configuration management plan of the SmartGuardian Project. This project aims to develop an application level web gateway which will be used by the network administrators of organisations in controlling the internet access of the computers in the Local Area Networks of those organisations. The configuration items (CI) which are individual items to which the configuration management plan will be applied are the three modules of the project, its submodules and related documents. The project is composed of three modules, namely the SmartGuardian Wall, Controller and the Crawler. The details of these modules and submodules are described in the coming sections. This configuration management plan will be applied to the implementation of these modules as a whole. In addition, this plan will also be applied to the test, debug and the documentation phases of the project.

Our project being a senior design project has many limitations in terms of time and human resources. The project is supposed to be completed in approximately three months. Since we are a team of five persons, this amounts to a fifteen man-months. We also have other considerations such as other courses and two of our team members are full time workers in other companies. Due to that our real workforce is less than the stated, but we are committed in completing the project on time.

We also have assumptions related with our project. All the team members are expected to finish their assigned work on time and get in touch with the group should a help is needed. The assignments are presented on the web page as a Gantt chart and living schedule which is updated every week.

In the next sections of the document the SCM management, development environment and tools, configuration items and control and the schedule will be presented.

SmartSoft Network Solutions, Inc.

1.3 Definition of Key Terms

CMB: Configuration Management Board

CI: Configuration Item

SCM: Software Configuration Management

1.4 References

IEEE Standard for Configuration Management Plans (IEEE Std 828-1998)

Roger Pressman, Software Engineering: A practioner's approach, Fifth Ed.

2 SCM Management

2.1 SCM Organization and Responsibilities

Since we are a small team composed of only five people, every person in the team has the responsibilities related with the configuration control. The team is called Change Management Board (CMB) and it includes all the team members. The director of the board is Tayfun, he will be responsible for the management and operation of the board. The purpose of the Change Management Board is ensuring the implementation of the plan as descibed in this document. The responsibilities of the board include:

- find bugs and report them to the group members via e-mail
- discuss on the changes and put up arguments against or in favour of these
- decide to commit the changes
- keep track of the changes made and reflect these in the future development

SmartSoft Network Solutions, Inc.

- update the configuration management plan if needed
- assign jobs to members

The board will be effective and in charge until the end of the project which is expected to be in the May 2006. The board will then discharge itself and dissolve.

The board respects the ideas that have sprung from the French Revolution. The authority and power is shared between all the members and the voting (balloting) mechanism will be used in making decisions. The director has the last word over the decision when the votes are counted equal (tie-breaking power).

2.2 Applicable Policies

All the software configuration management activities have to comply with the specifications presented in the course.

2.3 Development Environment and the Infrastructure

IDE: The C++ development environment to be used will be Anjuta. Anjuta is an integrated development environment (IDE) which uses tools provided by Gnome. In this respect it is small in size about 6 MB which is very small compared to Eclipse. It is also very easy to install and it uses less system resources. Our alternatives K-develop and Eclipse both had shortcomings. K-develop do not have advanced features and Eclipse uses too much resources. Eclipse is implemented in Java which is the reason for its inefficiency and slowness.

CVS: The development will be reflected to the CVS server as soon as possible, enabling collaboration between the team members. We were not able to use the CVS tool that comes with Anjuta so the CVS tool to be used will be LinCVS. We will only store the source files, configuration files and not the binary files to the CVS repository. The CVS folder structure will be created as follows: There will be separate folders for each main module which are the Wall, the Crawler and the Controller. Also shared sub modules which are used by more than one module will be stored in a different folder called “shared”.

Debugging: The debugging tool to be used will be Data Display Debugger (DDD) as it is an advanced visual tool easy to use.

Personnel: The workforce includes five persons in the team.

2.4 The Configuration Items

The configuration items consist of the modules and sub modules of the system and, their documentation. The configuration items and their descriptions are presented in the table below.

Item	Subitem	Description
Database (DB)		The system database
Database Communicator		The module that is an interface for the DB
Web Connector		The module tha will manage the connection.
HTML Parser		The module that will be used for

SmartSoft Network Solutions, Inc.

Item	Subitem	Description
		parsing and modifying the html code.
SG Wall		The main module that is responsible for filtering, blocking and logging.
	Authenticator	Used for authentication of the users (clients).
	Request Handler	The module for handling the requests of the clients.
	Verifier	This module will be responsible for filtering and blocking the requests and responses.
	Logger	A module that will log the requests and responses.
SG Controller		The web user interface for controlling and monitoring the system in general.
	Authenticator	The module for authentication of the administrator.
	Interface	This module will prepare the interface to be shown to the administrator and users.
	Reporter	This module will be used for monitoring the system.
	System Settings	This module will manage the system settings

Item	Subitem	Description
	Logger	A module that will log the administrative actions.
SG Crawler		The main module that will classify the websites according to their contents.
	Web Page Content Analyzer	This module will analyze the web pages and create statistical data.
	Web Page Classifier	Module for classifying a webpage using the statistical data according to the Bayesian Algorithm.
Documentation		
	Installation Manual	
	User Manual	
	Software Specification	
	Release Notes	

2.5 Configuration Control

2.5.1 Change

The primary method of communication between the members of the team is e-mail. Any change requests will be first proposed to the team using an email. The properties of the e-mail will be:

- The email subject should start with “Change Request” and continue with the module name the change is proposed to.

SmartSoft Network Solutions, Inc.

- The mail content should clearly include the change description, along with the reason and exact place in the code repository the change is requested.
- The main responsibility of the change is attributed to the main handler of that module, which is defined in the living schedule.

The changes will be discussed in the mail group first. If the change request is a major one it will be discussed in the following meeting. We will meet at least once on the weekends and one during the weekly meeting with the assistant. The change proposals will be discussed, voted and implemented if accepted. If a tie happens in the voting the final decision will be made by the CMB director.

The changes are going to be recorded in the source file header. The change number and the author name will be included, along with the date the change was performed.

2.5.2 Bugs

In the case of finding a bug, a different kind of methodology will be followed. The reporting e-mail will be similar to the change request with the following modifications:

- The email subject should start with “Bug” and continue with the module name and if known the class name.
- The mail content should include the inputs that have caused the malfunction or the environment (test-bed) and include the stack trace if an exception is thrown.
- The main responsibility of the bug is attributed to the main handler of that module, which is defined in the living schedule. In case the bug is a trivial one the person who has found the bug may wish to fix it and send an acknowledgement to the team.

2.6 Schedule

The important milestones that will affect the plan's implementation are listed below.

1. SG Wall Test & Debug	Sun 09/04/06
2. SG Controller Test & Debug	Sun 02/04/06
3. SG Crawler Test & Debug	Sun 26/03/06
4. Overall Test & Debug	Sun 16/04/06
5. Documentation Revision	Tue 16/05/06