WINSTONSOFT

Configuration Management Plan

ACCIPP / CLASSIM

Can HOŞGÖR *1395094*

Çağla ÇIĞ *1448554*

Elvan GÜLEN *1448687*

Nazif İlker ERÇİN *1395003*

1. Introduction

1.1 Purpose of CMP

For a software project, modifications and updates are inevitable parts of the development lifecycle. Design, development and implementation decisions might change at any time; and since most components of software are codependent on each other, change in one part most often requires changes in other parts. Today almost all medium to large size software are developed with team work rather than personal, so a change introduced by one member of the team affects other members of the team. Conversely, a problem introduced by one member of the team automatically becomes a problem of the whole team.

Therefore, in order to prevent bigger problems in the future, all changes and updates to the software system must be made extremely carefully. For this purpose, this Configuration Management Plan that defines the guidelines for identification, management, control and audit of the changes and updates was prepared.

1.2 Scope of Document

This document is prepared to handle the configuration management activities of the Classim Project. The activities discussed here are applicable to all documentation, source code development, software and hardware tools used, and any other process involved.

This document clearly explains the responsibilities the group members have. The process of identifying the necessary changes and updates, management of these changes, and informing the other team members of the changes are described. During the development of the project, some steps will be followed in order to avoid any conflict about the current status of the project. In other words, this report is intended for the group members in first place. Our supervisor and the instructor are among the other audience of the document.

1.3 Definitions, Acronyms and Abbreviations

- **CA** Configuration Auditing
- **CCB** Configuration Control Board
- **CI** Configuration Item
- **CM** Configuration Management
- **CMP** Configuration Management Plan
- **CVS** Concurrent Versioning System
- **SCM** Software Configuration Management
- **SCMP** Software Configuration Management Plan
- **SCR** System Change Request

1.4 Document References

1. IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998
2. Software Configuration Management – METU Computer Engineering CENG 492
3. http://en.wikipedia.org/wiki/Software_Configuration_Management

1.5 Document Overview

This document consists of six main parts, each of which is described briefly below:

- Introduction: The introduction part explains the purpose of preparing a CMP and the scope of this plan. In addition, the abbreviations and terms are defined as well as the references which have been used. The Organizations CM Framework: This part explains the responsibilities of the group members for accomplishing the CMP. The tools that are going to be used are included as well.
- The CM Process: The identification, management, and the auditing of the CIs are discussed here.
- Project Schedule – CM Milestones: The deadlines for the CM activities are given in this part.
- Project Resources: The resources that will be used for CM activities are described in this part.
- Plan Optimization: This section explains the methods that can be used to optimize the CMP.

2 The Organizations CM Framework

2.1 Organization

In order to implement SCM activities, organizational units should be described properly. Therefore, the organizational context is one of the most important issues for WinstonSoft. ACCIPP has a hierarchy of organizational units. On the top of this hierarchy, there is CCB unit, which handles the CM activities. Since the project is developed by only 4 people, each member of the team is considered as a member of CCB also. Other organizational units that are controlled by CCB are as follows:

1. Configuration Management Team (CMT)
2. Developing Team (DT)
3. Testing and Debugging Team (TDT)
4. Release Control Team (RCT)

These 4 units are in contact with each other and each team member participates in these units.

2.2 Responsibilities

The responsibilities of the units mentioned above are as follows:

2.2.1 Configuration Control Board (CCB)

i. Reviews SCRs offered by testing and debugging team and analyzes the effects of the
ii. changes,
iii. Accepts/rejects the SCRs,
iv. Holds audits,
v. Reduces the negative effects of changes made.

2.2.2 Configuration Management Team (CMT)

i. Creates and maintains CMP,
ii. Coordinates CMP activities and ensures that the activities are implemented on time,
iii. According to the works done, updates CM schedule,
iv. Reports the changes to other units.

2.2.3 Developing Team (DT)

i. Main responsibility of CMT is to implement the project source code.
ii. According to the SCRs, implements the mistaken parts again.

2.2.4 Testing and Debugging Team (TDT)

i. Main responsibility is testing and debugging the source code,
ii. According to the result of testing process, offers SCR.
iii. Release Control Team (RCT)
iv. Main responsibility is to control the releases of the project,
v. Creates baselines.

2.3 Tools and Infrastructure

WinstonSoft uses TortoiseCVS and Visual Studio as project resources

2.3.1 Tortoise CVS

TortoiseCVS is a CVS tool for Microsoft Windows released under the GNU General Public License. Unlike most CVS tools, it includes itself in Windows' shell by adding entries in the contextual menu of the file explorer; therefore it does not run in its own window. Moreover, it adds icons onto files and directories controlled by CVS, giving additional information to the user without having to run a full-scale stand-alone application.

Since the development environment is mainly Windows, it is decided that Tortoise CVS provides an adequate CVS solution for the project.

3 The CM Process

3.1 Identification

ACCIPP project can be identified as a combination of the following CIs: Software, Data, Documents and Baselines.

3.1.1 Software

Since ACCIPP consists of several modules for different accesses to data, namely ACCIPP Database, there will be code CIs in it. Decoder module, AutoSensing module and Output module and the sub modules of these modules will constitute all code CIs for ACCIPP. By this way, using several threads for each module, ACCIPP will process faster.

3.1.2 Data

Additional to the code CIs, ACCIPP also deals with a huge amount of data stored in its database. Instant messaging (MSMSG) information, Email (POP3, SMTP) files and its contents (attached files etc.), File Transfer (FTP) information and the logs related to all of the connections mentioned above. Besides offline pcap files can be considered as data for ACCIPP.

3.1.3 Documents

When developing ACCIPP, every designed module and any additional feature should be documented to be permanent and accessed easily. For this purpose, the following documents are prepared and published:

- Project Proposal
- Requirement Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan
- Weekly Progress Reports

3.1.4 Baselines

Baseline CIs are related to the milestones of ACCIPP and can be divided as followings: Requirements Analysis, Project Design, Project Prototype, CMP, Implementation, Testing and Documentation.

3.2 Management and Control

Source configuration during the implementation of ACCIPP is planned to be done by following steps considered below.

3.2.1 Change Request

When it is decided by a member of WinstonSoft to change the code of the project in any phase of the development, that member is supposed to send an e-mail to the project team

e-mail address, with the subject "CHANGE REQUEST: *Module Name*". A text file containing the information below should be attached to this e-mail message:

- ID of the request
- Date of the request
- Name of the team member that makes the request
- Configuration Item and (possible) sub-items of the request
- Module related to the request
- Priority of the request
- Assigned members to implement the request
- Detailed description of the change (Including the reason to make the change.)
- Version of the Code that the change to be applied.

3.2.2 Request Evaluation

Members of WinstonSoft will be informed about the requested change. The members will read the text file attached to the mail considered above in detail individually. The requested change will be taken into consideration and discussed in the closest group meeting. If the group members approve the request, the change will be implemented by the assigned group members. The assigned members can possibly be different than the members requested in the Change Request e-mail. The deadline will be stated in the group meeting. If the members disapprove the changes, the changes will not be applied, so there will be no change in the timing schedule. In each case, the member that requests the change will append the result of the evaluation at the end of the text file (the file that s/he attached to the CR email) and posts the new file to the group email address with the subject changed as *CREvaluation_Date_RequestID*.

3.2.3 Change Implementation

In case of the group members approving the change request the implementation of the change will be done. The assigned group members will make the changes within the specified deadline. The applied changes will be tested by the assigned members in case of any system crash. After the testing is completed, the changed files will be uploaded to the CVS repository supplied by the department. After the implementation is completed the member that requests the change will reflect the applied changes in the text file (attached to the email after the evaluation) and posts it to the group email with the subject modified as *CRImplementation_Date_RequestID*.

3.3 Configuration Status Accounting

The aim of Configuration Status Accounting (CSA) is to inform the group members about the modifications and updates of the project modules. By CSA, any modification in CIs can be followed easily by the group members and a well synchronized pattern can be formed.

As well as the group members, the project assistant and any other authorized supervisor will be observing the development progress.

For any modification a member who makes the change should follow the instructions that are listed below:

- Testing the changed part in case of any conflicts or system crashes.
- Updating the changed file in CVS.
- Adding simple comment about the modification while committing the file (CVS).
- Adding detailed information about why the modification is done for the group members to be aware of the changes and the progress.
- Storing the modifications according to their dates.
- Announcing new status via email, instant messaging services or phone.

3.4 Auditing

Software configuration audits are quite important for ACCIPP since it is aimed to serve a high quality final product to the possible customers. For the consistency and integrity of the modules with each other, auditing plays a significant role.

After modifications, auditing stage takes place and helps to control the changes for being sure about their accuracy for the functionalities of ACCIPP. Using audits, the group members understand the differences between baseline and current configurations.

For this project, the main audit that is planned to be applied will be functional configuration audits. This audit aims to evaluate the quality of the product with considering the accuracy of the CIs whether they provide the necessary functionality with the requirement specifications and the implementation is well formed.

In order to meet all the specifications and requirements, the changes and updates for every module will be audited. Moreover, it is planned to make peer reviews within the group members, the supervisor and the customer (Siemens) to decide whether a change and its effects are really necessary for the project.

After the implementation of all modules is finished, physical configuration audit is applied to the final version of ACCIPP. It affirms that the related modifications matches the all items and confirms that quality control, testing and management are well planned.

4 Project Schedules – CM Milestones

4.1 Project Schedule

A significant part of the living schedule has been determined and the deadlines have been stated by the supervisors and all the group members are going to follow this schedule and

work systematically. Also the weekly meetings will be helpful to control the status of the project for following the deadlines. All the group members will do the assigned task in time and this will be controlled by the group meetings and the weekly meetings with the supervisor. Moreover, in the project website, the living schedule will be updated in every week.

4.2 CM Milestones

The milestones of the project are shown below:

- CMP Submission: 09.03.2008
- Text Based Protocols Module Audit: 07.04.2008
- Binary Protocol Module Audit: 19.05.2008
- Final Version of ACCIPP: 13.06.2008

5 Project Resources

In order to apply the decided CMP and follow the CM activities, CVS will be used. Since the group consists of four people, it is important to work on the same source without inconsistencies. By the help of CVS, the changes and updates to the source are handled easily and new versions are produced as a result of these situations without losing the older versions. TortoiseCVS will be used as the CVS client in order to retrieve the source. Additionally, in the group's web page, all project documentations and a living schedule as a Gannt Chart will be kept to help us follow the CM activities easily. Moreover, since all team members will be responsible for keeping track of the changes and informing all other members, the members of WinstonSoft construct the human resources part of the project.

6 Plan Optimization

The CMP will be a guide for WinstonSoft in the development of ACCIPP. However, there can be some changes during the development process. In such cases, the group will make necessary optimization in the CMP and the progress. Communication between the team members is an important issue. WinstonSoft is a group of four members having weekly meetings with group members and with team supervisor. As a result, it will be easier to follow the changes and make necessary optimizations. All team members will be responsible for being aware of the updates and changes in the system and informing the other team members. For this purpose, team members may use e-mails, instant messaging services and phones for communication to keep track of changes and delays in the progress schedule and make the plan optimizations as quick as possible.