

***Configuration Management Report***  
***For***  
***Cloud-SOMS***

***Prepared by***

*Erkin Yavuz*  
*Gökçe Çözen*  
*Kazım Buğra Tombul*  
*Sercan Pehlivan*

Middle East Technical University  
Ankara

19 March 2011

|   |           |
|---|-----------|
| <b>1. Introduction .....</b>                        | <b>2</b>  |
| 1.1. Purpose of Configuration Management Plan ..... | 2         |
| 1.2. Scope of the Document .....                    | 2         |
| 1.3. Definitions, Acronyms and Abbreviations.....   | 3         |
| 1.4. Document References .....                      | 4         |
| 1.5. Document Overview .....                        | 4         |
| <b>2. The Organizations CM Framework .....</b>      | <b>5</b>  |
| 2.1. Organization .....                             | 5         |
| 2.2. Responsibilities .....                         | 5         |
| 2.3. Tools and Infrastructure .....                 | 6         |
| 2.3.1. Software Development:.....                   | 6         |
| 2.3.2. Version Control.....                         | 6         |
| 2.3.3. Project Management .....                     | 6         |
| 2.3.4. Libraries.....                               | 6         |
| <b>3. The CM Process .....</b>                      | <b>6</b>  |
| 3.1. Identification.....                            | 6         |
| 3.1.1. Source Code.....                             | 6         |
| 3.1.2. Data .....                                   | 7         |
| 3.1.3. Documentation .....                          | 7         |
| 3.2. Management and Control.....                    | 8         |
| 3.2.1. System Change Requests.....                  | 8         |
| 3.2.2. System Change Evaluation.....                | 8         |
| 3.2.3. System Change Implementation .....           | 8         |
| 3.3. Configuration Status Accounting .....          | 8         |
| 3.4. Auditing.....                                  | 9         |
| <b>4. Project Schedules - CM Milestones .....</b>   | <b>9</b>  |
| <b>5. Project Resources .....</b>                   | <b>10</b> |
| <b>6. Plan Optimization .....</b>                   | <b>10</b> |

## **1. Introduction**

### **1.1. Purpose of Configuration Management Plan**

Everything changes in time so as our project. Considering modules, classes and implementation of Cloud-SOMS change in time, making a configuration management is inevitable for stable development process. Cloud-SOMS consists of several different modules such as User Component, Organization Management Component, System Management Component and Social Integration Component, and four developer members of Cloud-SOMS are simultaneously contributing to enhance infrastructure and software. Changes and updates may occur to any type of project materials such as development plans, source code, and release versions during the development lifecycle of the Cloud-SOMS. It is important to realize that those changes and updates are vital for the project since, update of a program is inevitable for maintainability. Configuration management is beneficial for Cloud-SOMS team in order to provide strong communication among members together with improving quality of the product and to report changes within the members.

Configuration Management Plan (CMP) is prepared for handling situations caused by changes and updates. As a software developer it is important to realize that applying configuration management really increases quality of the product. Benefits of CMP include the stability of design and implementation of all components and subsystems so that interfaces between subsystems are reliable and accurate. CMP will facilitate and ensure that appropriate communications occur among members and all being potentially affected by the change when changes are needed. Participants will be informed about changes, and they will have opportunity to take part in discussion of the changes, scope and impact. After change being made is decided, participants can modify their subsystems design as needed.

In summary, CMP helps developers control and identify the causes and results of changes in a software development process so that these all can be used in the implementation of changes. Guidance of a CMP is essential for all software teams' communication and coordination on a project.

### **1.2. Scope of the Document**

The scope of this document is about the identification of configuration management plan for project Cloud-SOMS and its development team. Configuration management activities that will be applied throughout the development process of the project are presented and these activities being applicable during the development and maintenance phases of the project are explained.

The document contains methodologies and activities designed to handle the changes. Besides, versions of the product and reports about all changes are included. Since the project contains a medium number of components, some integrated libraries are being developed by four members of the project. The related audience for this document is Cloud-SOMS software team, our assistant and instructors.

### **1.3. Definitions, Acronyms and Abbreviations**

- SCR: System Change Request
- SCMP: Software Configuration Management Plan
- CI: Configuration Item
- CSA: Configuration Status Accounting
- CM: Configuration Management
- CCB: Configuration Control Board
- CMP: Configuration Management Plan
- SCM: Software Configuration Management

Configuration management (CM) is a field of management that focuses on establishing and maintaining consistency of performance of a system or a product and its functional and physical attributes with its requirements, designs, and operational information throughout its life. <sup>[5]</sup>

Configuration identification is the process of identifying the attributes that define every aspect of a configuration item. A configuration item (CI) is a product (hardware and/or software) that has an end-user purpose. These attributes are recorded in configuration documentation and base-lined. Base-lining an attribute forces formal configuration change control processes to be affected in the event that these attributes are changed.

Configuration status accounting (CSA) is the ability to record and report on the configuration baselines associated with each configuration item at any moment of time. The members can then ensure that the results of this verification process conform to their expectations and meet the needs of their particular components or subsystem.

Configuration audits are broken into functional and physical configuration audits. They occur either at delivery or at the moment of affecting the change. A functional configuration audit ensures that functional and performance attributes of a configuration item are achieved, while a physical configuration audit ensures that a configuration item is installed in accordance with the requirements of its detailed design documentation.

The Configuration Control functions of the CM process manage how changes will occur by ensuring organized communication among team members, such that the activities and functional responsibilities of one member are not hindered by unannounced or unanticipated changes made by another member. The implementation of configuration control typically requires a chartered governing panel and a well-defined process for Cloud-SOMS these are the Configuration Control Board and the Change Request process respectively.

#### 1.4. Document References

1. Pressman, Roger S. (2001). Software Engineering: A Practitioner's Approach, Fifth edition. New York, NY: McGraw-Hill
2. IEEE Standard for Software Configuration Management Plans (IEEE Std 828-2005)
3. A. Dogru, Component Oriented Software Engineering, The Atlas Publishing, Turkey, 2006.
4. Software Configuration Management?, The presentation prepared in METU Computer Engineering Department for the course CENG492 Spring 2010
5. [http://en.wikipedia.org/wiki/Configuration\\_management](http://en.wikipedia.org/wiki/Configuration_management)
6. <http://en.wikipedia.org/wiki/NetBeans>
7. [http://en.wikipedia.org/wiki/Apache\\_Subversion](http://en.wikipedia.org/wiki/Apache_Subversion)
8. <http://en.wikipedia.org/wiki/Trac>
9. [http://en.wikipedia.org/wiki/SSH\\_File\\_Transfer\\_Protocol](http://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol)

#### 1.5. Document Overview

**Introduction:** The need for CM, why configuration management is essential in project management, the aggregations that would be seen in this document and their descriptions, and finally the documents that guides us to understand CM, to develop our own configuration plan and to prepare this document are explained briefly in this part.

**The Organizations CM Framework:** First, the way how Cloud-SOMS would be organized to accomplish CMP, the tools being used and the need for these tools. Then, group members' responsibilities are introduced in this section.

**The CM Process:** The identification of the configuration items, the methodologies that going to be followed for CM, and the plan for the audits are sub-parts gathered under this title.

**Project Schedule CM Milestones:** The milestones for CM and the related deadlines are mentioned.

**Project Resources:** The project resources needed for CM is mainly in this section.

**Plan Optimization:** How CMP will be optimized and the needs for optimization are discussed in the Plan Optimization part.

## **2. The Organizations CM Framework**

### **2.1. Organization**

All Cloud-SOMS members' have equal rights during the project management process, and each member contributes to CM to develop a successful project. However, some sub-teams are arranged, as there are some management process steps and configuration management. Hence, each member of Cloud-SOMS is a member of CCB, which is responsible for:

#### **Development Team**

Development Team is mainly responsible for implementing the modules of Cloud-SOMS project and making the changes that are requested by Testing Team. This Team will also be responsible from the releases

#### **Testing Team**

Testing team will be responsible for two main purposes. These purposes as follows:

- To check whether the modules meet the requirements.
- To give feedback about the modules and want to change requests (CR) if necessary.

#### **Configuration Control Team**

Configuration Control Team will supervise all the activities of other teams. However, the main responsibilities of this group are to review SCRs, accept or reject SCRs and monitor SCR.

#### **Configuration Management Team**

Configuration Management Team is responsible for maintenance of the CM organization. Hence, this group will keep the Configuration Management Plan up-to-date.

### **2.2. Responsibilities**

Since each member of Cloud-SOMS is also a member of CCB all the members take the same responsibilities and these responsibilities are the following:

- Conforming to CM schedule.
- Commenting about changes before committing resources through SVN.
- Emailing people about SCR.

## 2.3. Tools and Infrastructure

### 2.3.1. Software Development:

**NetBeans:** The NetBeans Platform allows applications to be developed from a set of modular software components called modules. A famous integrated development environment that supports mainly JAVA and many other programming languages with additional features such as content assistant, code completion. <sup>[6]</sup>

### 2.3.2. Version Control

**SVN:** Subversion (SVN) is a version control system initiated in 2000 by CollabNet Inc. It is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS). Subversion (SVN) is a version control system that maintains current and historical versions of files such as source code, web pages, and documentation and stores these file in a server. NetBeans SVN plug-in to commit or update our source code is used. <sup>[7]</sup>

### 2.3.3. Project Management

**Trac:** Trac is an open source, web-based project management and bug-tracking tool. <sup>[8]</sup>

### 2.3.4. Libraries

**SFTP / Website:** The library files is going to be kept in Cloud SOMS's website since these files do not change during the implementation phase. Every member are going to reach these libraries through the website via Internet or SFTP to reduce the SVN work.

## 3. The CM Process

### 3.1. Identification

The CI's are an important part to identify Cloud-SOMS project. CI's are categorized into three parts, namely, source code, data and documents.

#### 3.1.1. Source Code

Coding phase is the most important part of the project and maintained by all the members of Cloud-SOMS. CI's of the source code are named according to its functionalities. Each component in Cloud-SOMS project is also a CI and they are explained below:

**User Component:** This component provides user management interface.

**Organization Administration Component:** This component provides organization management interface.

**System Administration Component:** This component provides system management interface.

**Facebook Component:** This component provides interface for Facebook integration.

**Twitter Component:** This component provides interface for Twitter integration.

### **3.1.2. Data**

Data consists of the main datastore. Datastore is a CI as a whole and needed for storing all information related to objects in Cloud-SOMS.

### **3.1.3. Documentation**

In the project, documentation is a mandatory CI. The documents created so far are the following:

#### **Reports**

- Project Proposal
- Requirements Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan

#### **Web Documents**

- Website
- Blog

#### **Development Process**

- Weekly Reports
- Blog Entries
- Twitter Updates

#### **User Manuals**

#### **Tutorials of End Product**



## **3.2. Management and Control**

### **3.2.1. System Change Requests**

Minor SCRs are directly added to the system and handled by SVN and requires no extra information. Major SCRs are controlled by the Trac system. In this system a SCR consists of:

- Team member name
- Description
- Date
- Deadline
- Related Module
- Priority
- Version

When a SCR is reported, a ticket that can be seen by all the team members, will be opened.

### **3.2.2. System Change Evaluation**

The discussions about evaluation of SCR are maintained on tickets in the Trac system. Also all SCR's will be discussed face to face in team meetings. During the evaluation, each member can stress his/her opinions freely and evaluation will be based on these opinions.

### **3.2.3. System Change Implementation**

If SCR is approved after an evaluation, all the possibly affected CI's will be detected and the related changes will be applied based on detected CI's. Then the version of the CI will be updated by committing the related resources through SVN.

## **3.3. Configuration Status Accounting**

Configuration items have been introduced in previous sections and the information about related configuration items are needed to be stored, since the control of configuration items gets harder as more changes are made. As keeping track of development process of the project is one of the most essential factors to be handled, different ways to express these changes and updates simultaneously are used. This informing process will not only include the intercommunication of the team but also external world that follows our project.

The mentioned information is made up of the following: configuration identifications, change requested information and information related to details of implementation. As project approaches to finish, information about the details of configuration items are kept in the

comments of SVN commits. Besides, meeting reports will be a guide through the common changes. Moreover, versions of the documents will be controlled via clearly explained comments. Also the SVN comments are especially written accepting the following points:

- Commits should not contain two unrelated updates or changes.
- Everyone t taken part in the commit should be mentioned in the logs of the commit.
- Commit comments should describe the change made briefly that everyone can understand.
- Related updates' commit to a ticket in Trac system should contain the ticket id in the log

Cloud-SOMS project members also use the team's blog in order to post entries related to development process and the status of the project. These blog posts may include tutorials, which express best-practice experiences of a member both to other group members and visitors. The blog is also accessible via the website of the team.

### **3.4. Auditing**

All the members of Cloud-SOMS will do auditing. During the auditing phase of SCR's, the changes that are made on a CI will be checked whether the changes are correct or not. In the weekly meetings, by using appropriate testing methods, auditing of the source code will be done. Auditing of the data will be done according to decisions made previously in the design phase. For example, number of concurrent users is an important criterion for our application so it should fulfill certain needs. Also, each developer has to check its own code so as to understand its code is working as intended. This is a personal responsibility of component developers and should be done before each commit into the SVN repository. Each commit should at least be compiled correctly. By this policy, the source code in the repository is always trusted to be compiled and working. Besides, these functional audits, members should observe the physical and process status each week. Project schedule should be checked and updated regularly in order to obey the timing constraints. Number of open tickets and resolved ones should also be evaluated in order to maintain sustainability.

## **4. Project Schedules - CM Milestones**

We have already prepared a living schedule, which has tasks and milestones. It can be shown at Cloud-SOMS website. Development has been divided in components so as to ease workload sharing. These separated tasks can also be found at the living schedule. Apart from these minimal divisions of force, the milestones of Cloud-SOMS are the followings:

- Delivery of CM: 20 March 2011
- First Development Snapshot, Demo: 8 April 2011
- First Release, Demo: 13 May 2011
- Final Release, Demo: 10 June 2011

## **5. Project Resources**

- SVN
- TRAC
- NetBeans
- SFTP
- Website

## **6. Plan Optimization**

CMP will be a guide for coordination and progress of Cloud-SOMS. Any updates or changes in CM schedule will be controlled by CCB. When an update or change occurs, all group members will follow this via TRAC. There will be meetings regularly to keep track of progress of configuration items. According to these meetings, the living schedule will be updated regularly. These meetings will keep us all up to stick with the schedule but as in all software development processes there can and will be some unpredicted problems. During the project, if there is a need to change in CM plan, Cloud-SOMS members will meet and adjust the schedule accordingly.