Team Members:
Taha Bekir Eren 1448646
Erdem Karahan 1347616
Mine Karakaya 1395136
1 Introduction
   1.1 Purpose of CMP
   1.2 Scope of Document
   1.3 Definitions, Acronyms and Abbreviations
   1.4 Document References
   1.5 Document Overview

2 The Organizations CM Framework
   2.1 Organization
   2.2 Responsibilities
   2.3 Tools & Infrastructure

3 The CM Process
   3.1 Identification
   3.2 Management and Control
      3.2.1 Change Request
      3.2.2 Evaluating Changes
      3.2.3 Approving or Disapproving Changes
      3.2.4 Implementing Changes
   3.3 Configuration Status Accounting
   3.4 Auditing
      3.4.1 Functional Auditing
      3.4.2 Process Auditing

4 Project Schedules - CM Milestones

5 Project Resources

6 Plan Optimization
1 Introduction

1.1 Purpose of CMP

As the components have started to be implemented; working on IMBO as a team of three people gets harder due to the fact that one minor change done by a person in a single component can affect the whole project at all points. The high possibility of required changes during the project and their effects on the project directly lead us to Software Configuration Management Plan. The purpose of our Configuration Management Plan is to lay down the identification, management, control, status accounting and auditing phases of our Configuration Management Process. Consequently; in conditions where changes are required, there will not be any problematic issues as all possible problems relating changes have already been eliminated during Configuration Management Plan process.

1.2 Scope of Document

The scope of this document has the main headings: Organization Configuration Management Framework, Configuration Management Process, Configuration Management Milestones, Project Resources and the techniques for Plan Optimization. Organization Configuration Management Framework gives information about the organization, the distribution of responsibilities according to the organization and our CM Tool choice.

1.3 Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMP</td>
<td>Configuration Management Plan</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>SCMP</td>
<td>Software Configuration Management Plan</td>
</tr>
<tr>
<td>CCB</td>
<td>Configuration Control Board</td>
</tr>
</tbody>
</table>
1.4 Document References

1- ) Configuration Management Document
   From Ceng 492 Presentation
2- ) Sample Configuration Management Plan
   From www.nasa.gov/centers/ivv/pdf/170879main_T2401.pdf
3- ) Configuration Management Plan
   From www.usdoj.gov/jmd/irm/lifecycle/appendixc7.htm
4- ) Software Engineering: A Practitioner’s Approach
   From Roger S. Pressman

1.5 Document Overview

Our SCMP is divided into six sections:

Introduction: In this part, the purpose of CMP and the scope of this document are denoted. In addition, acronyms, abbreviations and document references used in this document are stated.

The Organizations CM Framework: In this part, the organization is formed firstly. According to this organization, the responsibilities of each team member are determined. Moreover, the tool we will be using during the project is stated.

The CM Process: In this part, identification, management, control, status accounting and auditing phases of CM Process are described.

Project Schedules – CM Milestones: In this part, the milestones for
configuration management activities are given.

**Project Resources:** In this part, needed project resources for CM activities are given.

**Plan Optimization:** In this part, techniques enabling optimizations about our SCMP are described.

## 2 The Organizations CM Framework

### 2.1 Organization

Taha Bekir Eren is the configuration manager. He is responsible for tracking changes made to system and updating the project schedule when necessary. Since we have a 3 person team, all of the team members are considered as a member of Configuration Control Board.

There are four units controlled by CCB:

- Configuration Management Team
- Developing Team
- Testing and Debugging Team
- Release Control Team

All of the team members participate to each unit.

### 2.2 Responsibilities

Since all team members participate to all units, we are listing the responsibilities of the units as a single list. All team members are responsible for the below:
- Reviewing SCRs offered by TDT and analyzing the effects of the changes,

- Accepting/rejecting the SCRs,

- Holding audits,

- Creating and maintaining CMP,

- Coordinating CMP activities and ensuring that the activities are implemented on time,

- According to the works done, updating CM schedule,

- Reporting the changes to other members of the unit,

- Implement the project source code,

- Testing and debugging the source code,

- According to the result of testing process, offering SCR,

- Creating baselines.

  All of the above activities are supervised by project leader.

### 2.3 Tools & Infrastructure

We will use:

- SVN (Subversion) system provided by AGMLAB Company as our code repository. With the services of SVN, each team member can work on the source code in a parallel fashion, and get the latest changes made by other members in an incremental way. With SVN, a member can see the updated lines of source code from a creating revision. This feature makes bug tracking easier, because when a
new bug is introduced, it is probably result of latest changes to the source code. Also conflict resolution features of SVN makes teamwork on the same source code file very easy.

- Tortoise SVN, which is a graphical client for the SVN. Tortoise SVN provides almost every feature of SVN with a GUI. For example, one can see the latest changes applied to a source code file in a GUI environment. If a conflict which needs manual intervention to the source code occurs, developers can see the conflicting lines of code in a GUI.

3 The CM Process

3.1 Identification

Current state of our project consists of three items:

- Source Code
- Database
- Documentation

Source Code:

All source files those are necessary for our project to be compiled and deployed properly are included in this configuration item. These files consist of:

- Code files created and maintained by our team
- Library files such as .net framework dlls, or jar files
- WSDL files
- CSS files

- ASP.NET files (not code, just design)

- Configuration files (such as web.config file of an ASP.NET project)

These files will be managed by our SVN server.

**Database:**

Our project has to make extensive use of a DBMS to accomplish certain features. This requires a careful design and maintenance of database files. Database structure and data are separated from our standpoint. Database structure must have strict correspondence with our source code files to obtain a working system. At this point below entities should match between source code and database structure:

- Table name

- Field name

- Field type

- Field size

- Understanding of constraints

Handling the evolution of a database structure is harder compared to code files because a database file has a binary structure. If we try to maintain a binary copy of our database in the SVN repository, it will be very inefficient and resource consuming. Instead we will store database scripts in our SVN repository. Today most database vendors provide features that generate a corresponding sql script that will create the exact database schema from scratch. When a change is made to
our database schema by a team member, he/she will commit the resulting script to the repository.

We will also maintain scripts to populate tables with sample data for testing and debugging purposes. Unlike the database structure script, there may be several populating scripts to test and debug several scenarios.

**Documentation:**

Documentation consists of all documents created by our team for the IMBO project. These are design documents, progress documents, and schedule documents. All these documents are stored in our website and updated regularly. All team members add their weekly progress reports to the website. When necessary, project schedule and milestones are updated.

### 3.2 Management and Control

#### 3.2.1 Change Request

Since we are a small team, instead of a complicated bug tracking system, we will use our Google group for SCR communication. Any team member can issue a change request by filling following fields:

- Priority of the request
- Date and Time of the request
- Modules and files that will be affected by the possible change
- Reason of SCR
- Detailed explanation of SCR
This process is not necessary for every modification in the source code. Generally each team member implements a standalone module of the system. When a team member wishes to change a code piece which will not affect relations with other modules (like call interfaces) he/she may do without seeking for approval. These changes will be tracked and reviewed with the help of SVN.

3.2.2 Evaluating Changes

Normally project leader is authorized to evaluate SCRs. Requests are evaluated according to the current schedule of the project, accuracy of the request, priority of the request, and human resource issues. If the SCR is a complex one or it has complications, project leader may decide to call team members for a meeting in which SCR will be discussed. Or the issue may be discussed in the next regular meeting.

3.2.3 Approving or Disapproving Changes

It is up to the project leader to approve or disapprove the SCR. Project leader declares his/her decision in the meeting or the mail group.

3.2.4 Implementing Changes

Project leader assigns an ID number to the SCR. Implementation of the changes are done under the supervision of project leader by team members charged by him/her. When implementers of the change commit changes to the SVN repository, they have to issue the ID of the SCR in the description field. Following information will be stored by the SVN automatically:

- Revision number before and after the changes
- List of affected items
- Date and time of commitment
- Name of the contributing team member

### 3.3 Configuration Status Accounting

Below items will be tracked in mail group of the project:

- Change requests
- Builds
- Bug reports
- Release reports

Every two weeks a report will be prepared for each of the above items to identify the current status of the project.

### 3.4 Auditing

#### 3.4.1 Functional Auditing

We will use functional auditing to evaluate our product's quality. This audit will be held before the final release of our product.

#### 3.4.2 Process Auditing

Process auditing will be done during the regular meetings. Project leader ensures that defined process is strictly followed.
4 Project Schedules - CM Milestones

The Gantt-Chart; which will be updated regularly according to our implementations for the project; is our project schedule.

The CM Milestones are as follows:

- CMP Submission 09.03.2008
- Database Layer Module Audit 31.03.2008
- Jabber Plug-in Module Audit 05.04.2008
- Web Application Audit 03.05.2008

5 Project Resources

We will be using SVN; which enables simultaneous works of different people on the same modules without any incoordination. So, we will be able to reach every version of our implementations. Moreover, our web page will be satisfying documents and regularly updated Gantt-Chart of the project.

6 Plan Optimization

A member using SVN can make up newer versions of the implementations in which changes take place without destroying old versions without informing other team members; yet pursuing and updating the plan is the responsibility of each team member. Therefore decisions about updates will take place after the weekly meetings. If team leader or the team decides not to update the plan, newer versions containing changes in SVN will be deleted. Consequently each team member has the ability to work independently; but the coordination between team members is still maintained.