CONFIGURATION MANAGEMENT PLAN
## Contents

1. **INTRODUCTION**
   - Purpose of Configuration Management Plan ................................................................. 3
   - Scope of the Document ..................................................................................................... 3
   - Definitions, Acronyms and Abbreviations ........................................................................ 3
   - Document References ........................................................................................................ 4
   - Document Overview ........................................................................................................... 4

2. **THE ORGANIZATION OF CM FRAMEWORK**
   - Organization .......................................................................................................................... 5
   - Responsibilities ..................................................................................................................... 5
   - Tools & Infrastructure .......................................................................................................... 5

3. **THE CM PROCESS**
   - Identification ....................................................................................................................... 6
     - Source Code ...................................................................................................................... 6
     - Data .................................................................................................................................. 6
     - Documentation .................................................................................................................. 7
   - Configuration Management and Control ............................................................................ 7
     - Change Requests .............................................................................................................. 7
     - Evaluation of the Request ............................................................................................... 7
     - Approving Or Disapproving Changes ............................................................................. 7
     - Implementing ................................................................................................................... 8
   - Configuration Status Accounting ......................................................................................... 8
   - Auditing ............................................................................................................................... 8

4. **PROJECT SCHEDULES AND CM MILESTONES** .............................................................. 9

5. **PROJECT RESOURCES** .................................................................................................... 9

6. **PLAN OPTIMIZATION** ..................................................................................................... 9
1. INTRODUCTION

1.1. Purpose of Configuration Management Plan

MasterMind project is a complex gaming service which consists of making a commutation between patients and experts via game and web technologies. To prevent confusion among developers, our team aim is to use new technologies in order to manage the whole project. That is why, configure management plan (CMP) is an essential tool for huge projects. In addition, CMP helps the developers not to make crucial mistakes during developing the projects. CMP allows our team to integrate changes to the project easily, since it is supposed to reduce the side effects of the changes. Furthermore, it is for sure that if a member of the project makes a change, CMP will help the members to apply these changes without errors.

1.2. Scope of the Document

The specification of the configuration management plan for game project of MasterMind group is the scope of this document. Configuration management activities which will be applied during the development process of the software are outlined in this plan.

This plan specifically defines the organization of the team, responsibilities of team members and timeline which has been planned to achieve the CM process.

1.3. Definitions, Acronyms and Abbreviations

- CM Configuration Management
- CMP Configuration Management Plan
- CI Configuration Item
- CCT Configuration Control Team
- CSA Configuration Status Accounting
- SCR Specification Change Request
- SVN Subversion
- CMT Configuration Management Team
- SDT Software Development Team
- TT Testing Team
- RCT Release Control Team
- BCID Brain Computer Interface Devices
- Emotiv EPOC An EEG-based BCI device
1.4.  Document References


1.5.  Document Overview

This document is divided into six sections, the Introduction, the Organization, the Configuration Management Process, Project Schedule, Project Resources and Plan Optimization. The organization of this document is as follows:

- **Introduction**: This section consists of purpose and scope of the configuration management (CM), some definitions, acronyms and abbreviations, document references

- **The Organization of CM Framework**: This section includes organization and responsibilities of the project team on CM. In addition, it states the tools and infrastructure of the process.

- **The CM Process**: This section describes the configuration management process to be used

- **Project schedules – CM Milestones**: Important dates and milestones are listed according to the course syllabus.

- **Project Resources**: This section describes the resources of MasterMind group.

- **Plan Optimization**: This section explains methods which are going to be used to optimize CMP.
2. THE ORGANIZATION OF CM FRAMEWORK

2.1. Organization

CM activities are handled by the following groups.

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

- **Configuration Control Team:** Configuration Control Team will supervise all the activities of other groups. However, the main responsibilities of this group are reviewing SCRs, accepting or rejecting SCRs, monitoring SCRs.

- **Software Development Team:** The main task of this team is to implement all modules of the game. In addition, the responsibilities of this team include creating releases, updating project source code via SVN and fixing bugs.

- **Testing Team:** The main task of this team is to find bugs in the game. Furthermore, this team will create test cases and use cases, testing whether initial requirements are satisfied or not.

- **Release Control Team:** This team will monitor versions of the game. Moreover it is responsible for merging different branches of the game.

2.2. Responsibilities

Several modules are created based on their functionality. Through these modules workload is distributed to each member, since we are all in SDT, CMT, CCT, TT and RCT. In addition, if any changes occur in the source code it will be represented as comment via SVN.

2.3. Tools & Infrastructure

During the development and management of the game, the following tools will be used.

- **NetBeans**: Refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala and Clojure. We use NetBeans for developing all modules of the projects namely the Web Site, Mobile Application, Server since NetBeans offers a good environment for developing with these technologies such as JSP/Servlet, J2ME, Java.\(^1\)

---

• **SVN (Subversion)**\(^2\): In software development, Subversion (SVN) is a version-control system. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

• **Trac**\(^3\): Trac is an open source, web-based project management and bug-tracking tool. Trac allows hyperlinking information between a computer bug database, revision control and wiki content. It also serves as a web interface to a version control system, in our case SVN. With the help of the Trac, we will assign tasks to the members of SDT and also SCRs are raised via Trac to the CCB.

• **Emotiv SDK**\(^4\): The Emotiv Software Development Kits offers the opportunity for you to create applications that can be controlled by your mind and/or leverage the Emotiv EPOC technology to conduct EEG research. All of the Emotiv SDKs include a 14 channel, high resolution, neuro-signal acquisition and processing wireless neuroheadset and proprietary software toolkit that exposes APIs and detection libraries. The kit provides an effective development environment that integrates well with new and existing frameworks.\(^2\)

### 3. THE CM PROCESS

#### 3.1. Identification

We have determined the following Configuration Items (CI):

##### 3.1.1. Source Code

MasterMind project packages forms the Source Code CI. SVN is used to manage the source code. In addition, in order to share and store the source code at the same time SVN is preferred. For each module in the Mastermind project we have made different folders in the SVN, such as `../Mastermind/Web`, `../Mastermind/Game`, `../Mastermind/Server`.

##### 3.1.2. Data

Main database of MasterMind project is held on department’s MySQL server. There is a crucial correspondence between data in the game software and the main database.

---

\(^2\) [http://en.wikipedia.org/wiki/Subversion_(software)]
\(^3\) [http://en.wikipedia.org/wiki/Trac](http://en.wikipedia.org/wiki/Trac)
3.1.3. Documentation

We have prepared the following documentation up to now: Project Proposal, Requirement Analysis Report, Initial Design Report, Detailed Design Report, Revised Design Report, Configuration Management Plan. Group has a section on the website and each member of the group writes his weekly reports. In addition, we update the schedule of the project weekly. For the users of Mastermind game, User Manual and Test Plan will be prepared and stored in the website.

3.2. Configuration Management and Control

3.2.1. Change Requests

In the weekly meetings, a group member or our assistant can request a change in any tasks. These requests are evaluated and if it is approved by all the members, these are assigned in the Trac system with a ticket. The changes can be reviewed by using SVN.

3.2.2. Evaluation of the Request

Since all team members are also a member of CCT, all members can comment on the request independent of assigned person. If the changes effect the other modules, these must be evaluated by all members during weekly meetings to find an optimized solution. Change requests are assigned to group members considering baselines and each member of the team is responsible for these changes.

3.2.3. Approving Or Disapproving Changes

Since there is no leader of MasterMind project team, the decisions are approved or disapproved by the all members of the team.
3.2.4. Implementing

In the case of the approval of the request by majority of the members, the following steps are applied:

- The changed or modified CIs are analyzed
- Necessary changes in the source code are updated in SVN
- Unit testing is performed

3.3. Configuration Status Accounting

CSA becomes an important concept as the project advances and includes activities for recording and reporting the CIs of the project. To maintain coordination between members and modules in such huge project, configuration status of all versions of the project is needed. Because each member can access on all comments, he is aware of all changes. Thanks to SVN tool, as it is easy to keep all versions of the project and comments. It is significant to track all change requests, bug reports and release reports for Mastermind group in the development of project. Therefore, writing a weekly report is responsibility of each member and it is overseen strictly.

3.4. Auditing

MasterMind group has its own unique auditing which is an important part of CMP. Communication between members is done by weekly meetings, sms, e-mail, trac system. These communications ensures the auditing. Each weekly meeting and its content are organized in the previous weekly meeting. All tasks are assigned to members and reminded via trac system and weekly meetings. The requests posed at the meetings are evaluated and applied if approved. During meetings, any disagreement is solved with finding an optimized solution. New researches are assigned to members in case of short term unforeseen problems. In addition to weekly meetings, extra meetings are arranged according to baselines. Because all modules are dependent to each other, coding and testing are done by together. Moreover, MasterMind group has an effective communication in itself that eases the auditing as well.
4. PROJECT SCHEDULES AND CM MILESTONES

We have defined our milestones dates so that we splitted our modules into several tasks and we specified our living schedule according to these deadlines. In our website, we are updating our schedule weekly in terms of tasks which are finished.

The milestones which will be followed by the developers are as follows:

- First Development Snapshot Demo (Mar 29, 2011)
- Pre-first Release Prototype (May 2, 2011)
- First Release (May 10, 2011)
- Final Release (Jun 13, 2011)

5. PROJECT RESOURCES

The resources of the projects can be listed as follows:

- SVN (Subversion)
- Emotiv EPOC
- NetBeans
- Java
- Epoclib Library
- MySQL
- XML
- Apache Tomcat Server
- The documents that we have prepared (reports, living schedule, Gantt Chart)
- MasterMind group Web Site

6. PLAN OPTIMIZATION

During the development and progress of the project, CMP will guide us. As we defined our tasks in the living schedule, all members of the team will obey these tasks. However, we may need to make some changes in the deadlines of the living schedule during the development process. Therefore, our living schedule can tolerate these changes. In the team’s weekly meetings, we can decide the necessary changes and optimizations. These decisions will be updated in the living schedule, Gantt chart and CCT. Updated schedule can be reached via project web site.