

TABLE OF CONTENTS

1. INTRODUCTION	2
1.1. Purpose of the Configuration Management Plan	2
1.2. CoreAccess Overview	2
1.3. Scope of Configuration Management Plan	2
1.4. Definitions, Acronyms and Abbreviations	3
1.5. Document References	3
1.6. Document Overview	3
2. THE CM FRAMEWORK	4
2.1. Organization	4
2.2. Responsibilities	4
2.3. Tools and Infrastructure	5
3. THE CONFIGURATION MANAGEMENT PROCESS	5
3.1. Identification	5
3.1.1. Code	6
3.1.2. Data	6
3.1.3. Documentation	6
3.1.4. Baselines.....	6
3.2. Management and Control	6
3.2.1. Requesting changes	7
3.2.2. Evaluating changes.....	7
3.2.3. Approving or disapproving changes	7
3.2.4. Implementing changes.....	7
3.3. Configuration Status Accounting	7
3.4. Auditing.....	8
4. PROJECT SCHEDULE & CM MILESTONES	8
5. PROJECT RESOURCES.....	9
6. PLAN MAINTENANCE	9

1. INTRODUCTION

1.1. Purpose of the Configuration Management Plan

In every software project, some parts of the project tend to be changed and/or modified as the time passes and the things get complicated. No one can exactly know whether the work done is permanent or it is object to change. A software group can have very different members who think differently and make their jobs in different way. Therefore when an inconsistency occurs at any phase of a project, team members are gathered and try to fix that inconsistency. This is usually done by changing some parts of a project. Of course, a change to a specific part of a project can affect other parts of it. Thus, these changes should be handled appropriately by team members.

In this document, as the group of CoreTech, we covered the Software Configuration Management Plan (SCMP) which is mainly about handling version and change controls. In other words, for a proper software project, the changes and updates should be first identified, then controlled, managed and finally audited and all the members in the project group should be informed about the concerned change or update. This type of configuration management plan is a must for a proper software project.

1.2. CoreAccess Overview

Our project CoreAccess is a mobile GIS application for people who want to spend their time with social activities. These activities include cinema, theatre, sport, music activities. The position of the user will be determined and according to this information, the appropriate results will be returned to the user as text and visual map. In addition to these properties, CoreAccess will also provide convenient transportation alternatives to the user.

In the first semester, as the group of CoreTech, we first made a detailed requirement analysis and found out the user functionalities. After our requirement analysis, we started to make design and showed our design progress via our complete design report which includes several diagrams. We have also started implementation of our project at first semester. We showed several prototypes of our project to our teaching assistant Oral Dalay. After forming a successful and complete Configuration Management Plan, we will continue the implementation of CoreAccess with great effort.

1.3. Scope of Configuration Management Plan

The main scope of this document is to define and apply SCMP to our project CoreAccess to have a product stable and easy to maintain. Under the name of SCMP Report, this document mainly includes identifying and managing software configuration items, handling version controls, auditing the changes and informing all necessary data to group members.

In this document, we first mentioned about the possible organization and responsibilities of the product CoreAccess for the configuration management. We are aware of the importance of stable organization. After that, the tools and infrastructure about controlling the version and change took place in our document.

Our Configuration Management Process is described in mainly four subtopics. These are identification, management and control, configuration status accounting and auditing. This part of the project is very crucial. If we can manage to form a proper configuration management plan considering this process, we think we will prevent some possible problems beforehand.

1.4. Definitions, Acronyms and Abbreviations

These are some abbreviations in the SCMP:

CCB	Configuration Control Board
CM	Configuration Management
CI	Configuration Item
SCR	Software Change Request
SCMP	Software Configuration Management Plan
CA	CoreAccess
MR	Maintenance Request
CCA	Configuration Control Activity
CVS	Concurrent Versioning System
CSA	Configuration Status Accounting
FCA	Functional Configuration Audit
PCA	Physical Configuration Audit

1.5. Document References

We benefited from the following references while forming our SCMP:

1. IEEE Standard for Software Configuration Management Plans (IEEE Std 828-1998)
2. 2004 – 2005 METU CENG Senior Project Groups Documents, <http://senior.ceng.metu.edu.tr/2005>
3. “Software Configuration Management” Presentation prepared by METU Computer Engineering Department for the course CEng 492
4. Version Control, <http://www.tortoisecvs.org/>

1.6. Document Overview

According to the references we utilized to create this document, we divided our SCMP Report in six parts:

1. *Introduction*: At this part, the general purpose and main scope of the document is stated. Moreover, we gave the references and the definitions, acronyms, abbreviations we used throughout the document.
2. *The Configuration Management Framework*: The CM activities which will be done through the project are divided among the group members to make each member be aware of their responsibilities. Besides that, the tools and infrastructure are declared clearly.
3. *The Configuration Management Process*: At this part, the technical issues about CM are mentioned. These are mainly identification, management and audit of CIs.
4. *Project Schedule – Configuration Management Milestones*: The schedule of CM activities and the milestones for them is located to this part.
5. *Project Resources*: The necessary resources for CM activities are stated at this part.
6. *Plan Maintenance*: At this part, the CM activities to make the project stable and to maintain the bugs of the project are discussed.

2. THE CM FRAMEWORK

2.1. Organization

The structure of the organization of our project and the roles of the members are the followings:

- **Change Control Team(CCT)**
CCT is responsible for accepting or rejecting SCR coming from customers or testers.
- **CM Update Team(CmUT)**
CmUT is responsible for updating the CM schedule according to the works done and reporting the CM activities.
- **Testing Team (TT)**
TT is responsible for testing the releases of the project before it arrives to the customer. Moreover, TT sends SCRs of the product to the developers if it is necessary.
- **Development Team (DT)**
DT is responsible for implementing the product and reimplementing the erroneous part of the project.
- **Version Control Team (VCT)**
VCT is responsible for following the version of the product.
- **Release Control Team (RCT)**
RCT is responsible for controlling the releases of the project.

2.2. Responsibilities

All individuals are in CCB and have to follow the schedule. The followings are the responsibilities of each member of the team:

- Perform the assigned CM activities.
- Inform all members when a change is done in CVS.
- Finish the jobs until deadlines.
- Attach comment after changed codes in the CVS.

2.3. Tools and Infrastructure

We are going to use CVS as our version control system. CVS server is supplied by our department. It is one of the most popular version control systems available today and it is open source. CVS keeps a central repository that includes current source code, past versions of the system, and logs that document changes to the system. CVS supports both text and binary files, but it is more efficient for text files. This is good for source codes; being able to follow the changes that are made to source code by other developers is very critical for the success of the project. Also, any previous version of a file can be seen by the help of CVS.

In the server, we will have 2 main directories; one for the PDA module and the other for the server side development of our project. Each will have its own tree structure. For the time being, we have already committed our PDA module to CVS server.

We prefer to use 2 main applications as CVS clients since using pure CVS is unnecessarily boring. Every group member will access CVS server from his/her own computer by the help of these tools. The first one is TortoiseCVS. It allows us to work with files under CVS directly from Windows Explorer and it is freely available under the GPL. With TortoiseCVS, we can directly check out modules, update, commit and see differences by right clicking on files and folders within Explorer.

Our second CVS client is in fact a fully integrated development environment for Java, namely Eclipse. We are developing the server side of our product with Java and Eclipse is a very preferable CVS tool which works under both Linux and Windows.

3. THE CONFIGURATION MANAGEMENT PROCESS

3.1. Identification

In order to identify the CoreAccess project, the CIs of the project is mentioned. These CIs are explained in following:

3.1.1. Code

Code CI consists of all implemented modules of the project. These modules are separated into two. One of them is client side code which means that this code is run in the PDA part. The other one is server side code which is run on the server.

3.1.2. Data

Data CI consists of the necessary map data and activity information. The map data is necessary for GIS manipulation. The activity information provides the user to search for different activities.

3.1.3. Documentation

In order to clarify the project mechanism, we have to document every phase of the project. These documents are as followings:

- Project Proposal
- Requirement Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan
- Test Specifications

3.1.4. Baselines

Baselines CIs mention the milestones of the project. These CIs are as the followings:

- Analysis of Requirements
- Detailed Design
- Developing Prototype
- SCMP
- Implementation
- Testing
- Documentation
- User Manual
- Installation Plan
- Executables

3.2. Management and Control

In this part of the CMP, configuration control activities of the CA request, evaluate, approve or disapprove, and implement changes to baselines CIs. These activities are explained in following:

3.2.1. Requesting changes

If a change and modification is necessary for a module, we can request one of the team members to fix it. We accept only our team members and our instructors change request during implementation. We can accept the customer MR after the release. In SCRs, we will use a template for mailing. This template consists of:

- Id number of the SCR
- Date of SCR
- Deadline of SCR
- Priority of SCR(1 to 4, 1 is highest priority, 4 is lowest priority)
- Owner of SCR
- Assigned member of SCR
- Description of SCR
- Change requested Module

3.2.2. Evaluating changes

According to deadline of the tasks and milestones, each member is responsible for determining the affects modules when s/he changes something in his/her module. Therefore, team members have to know the function all modules of the project even if they do not know the exact content of the modules. Moreover, when a SCR is created, the priority should be determined according to the milestone. As it is mentioned before, there are 4 types of priority. 1 is the highest priority and 4 is lowest one. If these rules are obeyed, it is easy to fix the SCRs.

3.2.3. Approving or disapproving changes

The SCRs are approved or disapproved by the CCB. Since all members of the team are CCB, mailing technique is used for this CCA. If one of the members hesitates the effect SCR, s/he can express his/her complain. After discussion, the CCB members decide to approve the SCR or not.

3.2.4. Implementing changes

After the SCR is accepted, the task assigned person is determined by the SCR creator as it is mentioned before. Then, the responsible person fixes the code according to the description and uploads it to the CVS. Thus, the others member can achieve the latest updated version of the module.

3.3. Configuration Status Accounting

Configuration Status Accounting mainly includes the activities for recording and reporting the CIs of the project. By CSA, we will report to all the group members about the status of the project. In addition to that, our assistant and our teachers will be informed. The related changes used to be informed by meeting reports and weekly meetings at the first semester. At this term, besides meeting reports and weekly meetings, we will use CVS which will include the latest changes and our website to manage the informing business. When a member puts

his/her source code to CVS, the related document or comments should be provided by him/her. Moreover, we still carry on the communication by the e – mail among group members and our assistant and the yahoo group for the Mobile GIS Implementation groups.

When a change recorded, we need some information which must be present for every change. First of all, the description of the change should be stated. This includes the information about why the change is needed and how the change is done. Moreover, we have to observe how this change affects the whole project and check whether there can be an inconsistency or not. Next, the people responsible for the change should be kept. When the things go wrong, the member who made the change knows much anybody else in the group. Finally, the changes must be numbered regularly as versions and the dates must be recorded.

3.4. Auditing

Software Configuration Auditing is an important part of Configuration Management which can let us produce a software product which has a high quality and meets the demands of the user. By proper auditing, we have the chance of observing whether changes are done correctly, the necessary information about the changes are done and all the related people are informed about the change process or not.

Our Software Configuration Auditing is mainly composed of Physical Configuration Audit and Functional Configuration Audit. While PCA is conducted after the final release, we think we apply FCA after releasing each new version. The main purpose of the FCA is to check if we can able to complete Configuration Management Development successfully and verify if our functional design baselines are convenient with our changes. On the other hand, the purpose of the PCA is verifying every system CI in the final release matches to prior specification of it. Moreover, the technical documentation and software product adaptation is confirmed.

4. PROJECT SCHEDULE & CM MILESTONES

We made a living schedule in the beginning of the semester which indicates the deadline of the tasks of the project. All members have to follow the regulations written in the living schedule. The schedule is on the website and it is updated regularly. Besides these responsibilities of team members, group has to make a meeting among each other once a week and with the instructor again once a week and should report the weekly process. Moreover, there are CM milestones of the project. These milestones are the followings with the deadlines:

- GPS module implementation – 04.03.2006
- PDA module implementation – 29.03.2006
- Content manager module implementation – 15.03.2006
- Transportation module implementation – 10.04.2006
- Server side implementation – 10.04.2006
- Database module implementation – 15.03.2006

- Integration of modules – 18.05.2006
- Testing – 25.05.2006
- Documentation support – 20.05.2006

5. PROJECT RESOURCES

In order to sustain CM activities, we need software tools, techniques, equipment, personnel and training. Our personnel only consist of developers and they are the most important resources of our project. All group members are responsible for the development and management of the project.

The main tool that will be used for CM is CVS. CVS is a must for projects which are being developed by several developers. Excluding the extreme cases, developers can independently or simultaneously work on the same source codes without any confliction. The changes are kept and previous versions of the files can be reached easily. CVS is helpful for our instructors also; they will be able to keep our development progress. We will use 2 CVS clients; TortoiseCVS and Eclipse. The details of these tools are explained in the Tools and Infrastructure header.

Finally, our web page is one of the CM sources since it includes all the project documents and living schedule which is being updated continuously. All of these are important resources to complete the project without any problems.

6. PLAN MAINTENANCE

Since we are group of 5 people, it is easy to follow the changes and CMP. We are all responsible for monitoring and maintaining CMP. During the weekly meetings we can decide the changes in the project and according to the decisions, we will update the CMP weekly if it is necessary. Moreover, CVS is helpful for checking in and checking out without confusion. Furthermore, the mail traffic among the members guides them about the changes in the project.