

**Middle East Technical University  
Department of Computer Engineering**



**ClimbPlanner  
Configuration Management  
Plan**

By

**Quattro Group**

*Authors:*

*Gökçen Güner*

*Mehmet Aktaş*

*Mehmet Uygur Akgül*

*Bilen Yavuz*

*Assistant:*

*Ali Orkan Bayer*

*09.03.2009*

1. Introduction	2
1.1 Purpose of CMP	2
1.2 Scope of Document	2
1.3 Acronyms and Abbreviations	2
1.4 References (Related Documents)	3
2. The Organizations CM Framework	3
2.1 Team Structure	3
2.2 Responsibilities	3
2.3 Tools & Infrastructure	4
3. CM Process	5
3.1 Identification	5
3.1.1 Engines & Sub-Modules	5
3.1.2 Documentation	6
3.1.3 Baselines	6
3.2. Configuration Management and Control	6
3.2.1. Development	6
3.2.2. Deployment	7
3.2.3. Change Request	7
3.2.4. Defect Tracking	7
3.3. Configuration Status Accounting	8
3.4 Configuration Auditing	8
4. Project Schedule	9
5. Project Resources	9
6. Plan Optimization	9

# 1. Introduction

## 1.1. Purpose of CMP

Software development is not an individual process. It requires a group of professionals to come together and identify problems, propose solutions and implement them. Together with these stages, testing and verifying the modifications, new approaches to existing or new problems is a big matter in development cycle of software projects. Configuration Management Plan steps in this point. Generally CM Plan of a project is developed to define, document, control, implement and audit changes to the various components of related project. The main purpose of this report is defining a CM Plan for CLIMBPLAN Project which is being designed by four people.

## 1.2. Scope of Document

This document is designed by members of Quattro Group and defines Software Configuration Management Plan. The activities discussed here are applicable to all documentation, source code development, software tools used, and any other process involved. This document clearly explains the responsibilities the group members have. This document is firstly designed for our group as guidance. Among the audiences there are our supervisor Ali Orkan Bayer and instructors Onur Tolga Şehitoğlu and Pınar Şenkul.

## 1.3. Acronyms and Abbreviations

<b>SVN</b>	Version Control System
<b>CI</b>	Configuration Item
<b>CM</b>	Configuration Management
<b>CMP</b>	Configuration Management Plan
<b>SCM</b>	Software Configuration Management
<b>GUI</b>	Graphical User Interface

## **1.4. References (Related Documents)**

- CLIMBPLAN Project Software Requirement Analysis Report
- CLIMBPLAN Project Final Design Report
- 2006 – 2007 METU CENG Senior Project Groups Documents, <http://senior.ceng.metu.edu.tr/2007>

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

### **2.1. Team Structure**

Our project group Quattro consists of four people. In previous term we have three people in our group. These 3 people have nearly same backgrounds and professions. In previous term we have no team leader and we choose decentralized team type. This term we have a newcomer to our group and this member has a big experience in software development process. As a result we choose the newcomer -Bilen Yavuz- as a group leader. Although we have a team leader, we don't have strictly defined roles in our group and it makes our group Controlled Decentralized. We have a good communication in the group and besides that we have weekly meetings. In these meetings we discuss the progress of the project, new problems and solutions to them and how to implement them. Like in every software development team, we divided tasks to members. All the members should attend the meetings; involve the development phases and testing phases of system. We have development team, testing team and documentation team. Because we have 4 members all members have roles in different groups.

### **2.2. Responsibilities**

As a controlled decentralized group, we have some restrictions that all of the members should obey and some roles for subgroups that the members of these sub-groups should obey. Firstly, general obligations for each member of the group are defined as:

- To comment on the code properly and explain the changes done efficiently while committing to SVN.
- To obey CM and Living schedule.
- Inform the other group members in a case of modifications and give logical explanations for these modifications.

- To attend weekly supervisor meetings and group meetings and explain the progress of the part he owns.

The team leader has some responsibilities like;

- Assigning tasks to members and controlling the progress of the sub parts of the system.
- Organizing intra-group communication and taking steps in a case of problem.

Development and testing teams have some responsibilities like informing documentation team and giving explanatory comments about the improvements about the project. Documentation team must be in communication with the other teams to collect related data about project.

### **2.3. Tools and Infrastructure**

Quattro Group has started to develop sub-modules of the CLIMBPLAN Project and it is still in progress. The size of the project group doesn't matter when it comes to concurrent code developing. Subversion (SVN) eases the development of code and also supplies versioning system as it is obvious from the name of program. SVN accounts are provided by Computer Engineering Department for all the members of CLIMBPLAN Project and we will use it to facilitate code development process. We will use Visual Studio for code development and it is well integrated to SVN.

Also versioning system is a good feature for developers and supervisors to see modifications, version differences and tracking the development process. Main steps involving in usage of general SVN applications are:

- Check out CI that is to be edited
- Edit CI
- Commit updated CI
- Check in the updated CI

As stated before, these steps must be taken carefully and while editing CI, the editor must clearly comment about the changes and their functionalities.

## 3. CM Process

### 3.1. Identification

We have divided our project to 3 CIs that can be managed and versioned separately, namely Engines & Sub-Modules, Documents and Baselines.

#### 3.1.1. Engines & Sub-Modules

- *User Interaction Engine*
  - Data Input Sub-Module
    - Responsible for gathering data from user
  - 3D Visualization Sub-Module
    - Visualizing map data
  - User Authentication Sub-Module
    - Authenticate the user with username/password
  - Simulation and Reporting Sub-Module
    - Simulate the map data with route and report the plan
- *Planning Engine*
  - Route Planner Sub-Module
    - Plan the route considering user data and map data
  - Comparison and Qualification Sub-Module
    - Give the result according to equipment and user restrictions
- *Simulation and Reporting Engine*
  - Simulation Sub-Module
    - Calculate the simulation process
  - Reporting Sub-Module
    - Send report data to printer or save it
- *Web Engine*
  - Weather Sub-Module
    - Gather weather forecast for the activity days
  - Map Sub-Module
    - Gather map data from web for Route Planning

### **3.1.2. Documentation**

- *Design Report*
- *Configuration Management Plan*
- *Installation Manual*
- *User Manual*
- *Living Schedule*
- *Testing Specification Sheet*

### **3.1.3. Baselines**

The baselines that are defined by Department and by the project group are very important in designing stages of the project. Project group will obey the baselines and members of subgroups will finish their tasks before the deadlines.

Some of the baselines listed below:

- First snapshot demonstration and development
- First release and test specifications
- Presentation
- Final Demonstration

## **3.2. Configuration Management and Control**

We need a main frame that describes the update process of the code and documentation for the project. This update process can be accomplished via four steps: Development, Deployment, Change Request, and Defect Tracking.

### **3.2.1. Development**

Development has the biggest role in the project design process. We thought carefully about the stages and approaches of development phase. Quattro group has 4 members and we divided these members to different areas like development, testing and documentation generally. In development area again members are divided to subgroups those are responsible for developing different sub modules. Each member has different roles in different subgroups. Through that method every member in the group will be aware of the each step taken in project development and nobody does more work than another. Also in documentation task although we have documentation subgroup, each member will report the modifications or problems about the part that he owns and in this way documentation subgroup's time will be utilized. Testing of an engine will be done by 2 people called as testing group, and this couple will consist of 1

developer and 1 another member who isn't involved in development of the related engine. In this way the testing group will be more effective and creativity encouraged.

### **3.2.2. Deployment**

Like every other business solution, the main intention of developing software is to answer all the needs of the customer. In previous term, we worked on the needs of users (in our case mountaineers), prepared a requirements analysis report and now we know the customer needs. The problem here is mainly in the implementation area. We know the needs, we know the solution but there is a huge gap to be filled between them. For an efficient deployment in each step of development, we will take user needs in to consideration and act accordingly. Because the success of software stands on how many of the customer requirements answered by the software, Quattro group's main intention is an efficient deployment that answers the customer needs as much as possible.

### **3.2.3. Change Request**

The main reason of requesting a change instead of directly modifying it, easing the development of the code and preventing some inconsistencies across the code. There are 3 steps for a change to be implemented over the code.

- Firstly a member requests the change on a module or engine. It doesn't matter he proposes the change on a weekly meeting or via e-mail, the member explains the reasons of change, possibly required changes on other sub-modules for consistency, and profits/costs of the change.
- The group discusses the change and considering profit/cost rate they refuse or accept the change. If the problem is an important one then group works on a suitable solution in the case of rejecting proposed solution.
- In the case of acceptance, the change will be committed and logged in the history of the current version.

### **3.2.4 Defect Tracking**

Defect tracking is another issue to be considered in Configuration Management. From the beginning we thought and designed defect tracking as testing stage. After the sub-modules finished the first testing stage begins. The reason of these tests is to be sure whether components work correctly. After that, second phase of testing begins and checks whether components communicate correctly. Following the tests that check the functionalities of engines, integration



tests begin and system integrity is checked. In each step user needs are taken into consideration and if there is problem between system features and customer needs then developers will try to understand and correct the situation. Also if there is a problem about working mechanisms of engines or sub-modules, then testers report it and developers try to correct the problem.

### **3.3. Configuration Status Accounting**

During the development of the CLIMBPLAN Project, we will provide some notes about status updates that are applied on the project. There will be Change Request Reports, Build Reports, Defect Reports, and Release Reports. Change Request Reports will be included in SVN repository but we don't mess these reports with small change requests. In these reports there will be important change requests that affect the course of project and the details like owner of the request, reasons for the change and results of the change. Build reports will be available in SVN repository right after build process for each engine is completed. In a case of problem during build process, if the problem is small and easy to solve then we will not include this problem in Defect Reports. For each stage (sub-module design, sub-module integration, engine integration), there will be a connected Defect Report, Change Request Report. Also release reports for alpha release, beta release and finale release will be available after they are completed.

### **3.4. Configuration Auditing**

Auditing is the act of evaluating an organization, system, process, project or product. As a controlled decentralized project group, we don't have a member for the role CM Manager. We are planning to audit the progress of the project in two ways. Firstly in weekly meetings and secondly in requested meetings.

In weekly meetings, all the members explain the progress about the parts they are responsible for. They state the difficulties they faced with, the problems occur during implementation or testing and propose solutions for these problems. Also new approaches to existing problems discussed here. Besides these update discussions, the group has auto control system and checks the progress, considering CMP and Living Schedule. If the progress of the some parts left behind the schedule, then whole group will try to help the sub-group which has left behind the schedule. At the end of weekly meetings, the plan for the next week discussed and some decisions made about the course of project.

In case of important issues like releases, demo preparations and for each version a total auditing will be done by the group. During these audition sessions if the group notice a problem

or needed improvement then these points will be added to audit reports and group will assign related task to a sub-group. After audition sessions, decided changes are made by assigned members and these members inform other members about the modifications over the project. These changes are traced on the audition reports so that other members can follow the progress of the results of the audition sessions.

#### **4. Project Schedule**

In previous term Quattro group consists of 3 people and we have scheduled tasks for these 3 people. This term we have a new member and we updated the task divisions according to current status. The task assignments are available in Living Schedule sheet in our web site:

<http://senior.ceng.metu.edu.tr/2009/quattro/>

The living schedule of the project will be updated in each week properly and progress rates for each task can be observed. Also we have some deadlines that some tasks must be completed before each deadline. The deadlines for defined tasks are here:

- Mar 12            First Development Snapshot Demo
- Apr 23            Test Specification
- May 7             First Release, Demo
- Jun 11            Final Package
- Jun 15/16        Final Demo

#### **5. Project Resources**

The basic application we will use in concurrent code development is SVN. Because we have 4 member in our group, preventing inconsistencies across the code is vital. The features of SVN provide members of group to develop code separately and commit them to create a whole and consistent project. We will use Visual Studio for code developing. MSSQL Server for holding user and group data. Windows Presentation Foundation (WPF) will be used for developing user interfaces. Also we will use Smart Draw and MS Word for documentation issues.

#### **6. Plan Optimization**

This CMP Report is a mainframe for development of CLIMBPLAN Project for Quattro Group. The progress will be compatible with CMP. Although we draw a mainframe in this CMP,

definitely we will face with problems during development and update CM according to new situations.

Throughout the project development, we will need optimization of CMP. Considering our weekly meetings and continuous communication with various tools, we will update our plan continuously. After every milestone, we must prevent our CMP becoming out-of-date.

The whole group has intercommunication and this provides us to notice and solve the problems very quickly. The decided changes over the design or code are done via SVN by an editor member and the members except editor will be informed about the changes and results immediately. This communication traffic ensures CMP is to be up-to-date. Also weekly meetings in group and weekly meetings between supervisor and group is a significant factor in updating and optimizing CMP.