



# **MAÇA YAZILIM**

**ONLINE VIRTUAL TEAM COLLABORATION PLATFORM  
WITH 3D GRAPHICS**



**CENG 491**

**Configuration Management Plan**

**METU**

**2007**

# Table of Contents

1. INTRODUCTION .....	3
1.1 Purpose of the Document.....	3
1.2 Scope of the Document.....	3
1.3 Definitions, Acronyms and Abbreviations.....	3
1.4 Document References.....	4
1.5 Document Overview .....	4
2. THE ORGANIZATIONS CM FRAMEWORK.....	4
2.1 Organization.....	4
2.2 Responsibilities.....	5
2.3 Tools & Infrastructure.....	6
3. THE CM PROCESS .....	6
3.1 Identification.....	6
3.1.1 Engines and Modules.....	6
3.1.2 Data.....	7
3.1.3 Documents.....	7
3.2 Management and Control.....	7
3.2.1 Engineering.....	7
3.2.2 Development.....	7
3.2.3 System Management.....	8
3.2.4 Change Request.....	8
3.2.5 Build and Deployment.....	9
3.2.6 Defect Tracking .....	9
3.3 Configuration Status Accounting.....	10
3.4 Auditing.....	10
4. PROJECT SCHEDULES – CM MILESTONES .....	11
5. PROJECT RESOURCES.....	11
6. PLAN OPTIMIZATION.....	12



## 1. INTRODUCTION

### 1.1 Purpose of the Document

Configuration management plan is an essential part of software development process. For the management and maintenance of the software, developers should adopt an evolutionary approach. During the implementation phase, many points of the project which were planned in the design phase will be reconsidered and updated. But even a minor change in the project can lead major consistency problems if there is not a well-organized configuration management plan. Defining certain procedures to follow in these kinds of problems helps the software protection against change related risks. That is the main function of configuration management plan.

### 1.2 Scope of the Document

The document introduces the configuration management plan of Maca Yazilim's senior project. It defines a methodology which the members are responsible of behaving accordingly and the standards of the project. The procedure to be followed when a need for modification occurs is provided in the document.

The intended audience for this document is the members of Maca Yazilim, the project assistant and the instructor.

### 1.3 Definitions, Acronyms and Abbreviations

CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CVS	Concurrent Versioning System
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
IEEE	Institute of Electrical and Electronics Engineers
CCB	Configuration Control Board



SCR	Software Change Request
TT	Testing Team
FTR	Formal Technical Review

#### 1.4 Document References

- Maca Yazilim Requirement Analysis Report
- Maca Yazilim Detailed Design Report
- *Software Configuration Management* slides, METU Computer Engineering – CENG 492, Spring 2004
- IEEE Standards for SCMP (IEEE Std 828-1998)

#### 1.5 Document Overview

The main subparts of the document are listed below:

- 1) *Introduction*: As the title implies, this section introduces the function and content of the CMP. Meanings of the abbreviations and the references used are also mentioned.
- 2) *The Organizations CM Framework*: The section describes the organization of the team, the responsibilities of team members and tools used during the project development.
- 3) *The CM Process*: The section
  - explains how to identify the current state of CIs
  - describes the methodologies followed for CM
  - defines the status updates to be provided and audit plans
- 4) *Project Schedule – CM Milestones*: Time plans are given in this section.
- 5) *Project Resources*: The resources needed during CMP are explained in this section.
- 6) *Plan Optimization*: The section covers the methods to be followed for optimizing CMP

## 2. THE ORGANIZATIONS CM FRAMEWORK

This section is about the organizational structure of the Maca Yazilim – SpadeShip Project. Organization, responsibilities, tools and infrastructure are explained below.

### 2.1 Organization

The organizational units that handles the SCM activities as follows:



- *Configuration Control Board (CCB)*

Since Maca Yazilim Team consists of four people which is a small number, every team member is involved in CCB. CCB approves or rejects an SCR, updates the CM schedule, holds audits. Generally CCB supervises every CM activity.

- *Testing Team*

Every member of the Maca Yazilim is also a member of SpadeShip Project's testing team. Testing Team generates SCRs after testing process and also team members ensure the implementation of SCR's.

Both case testing and unit testing are performed by TT members to ensure the functional capability and the reliability of every single part in the software.

- *Developer Team*

Developer Team also covers the whole team. They implement the change requests prepared by the testing team if the requests are approved by CCB. DT implements the modules, creates the snapshots and releases. They also establish baselines.

Since DT members are direct originators and reviewers of the code base, they may also make change requests.

## **2.2 Responsibilities**

Each team member is also a member of CCB, test team and development team so each of them has the same responsibilities;

- Identify configuration items as CCB member
- Inform the other members about generated SCRs at the meetings or by via e-mail as a tester
- Approve or reject an SCR as CCB member
- Comment the developed functions and files as developer
- Upload the updated files to CVS as developer
- Ensure the SCRs are properly implemented as tester
- Update CM schedule as CCB member
- Keep up with the schedule as developer



## 2.3 Tools & Infrastructure

Tools that are chosen for configuration management are;

- **Tortoise CVS**

CVS (Concurrent Version System) is used for version management of the project. CVS gives the opportunity that each developer can reach the most current version and all the version history of the project. Tortoise CVS is a tool that enables to work with files under CVS version control directly from Windows Explorer.

- **SSH**

SSH protocol is also used for reaching the source files and documents under CVS.

- **Microsoft Visual Studio**

Microsoft Visual Studio is the development environment for the project. Since the simulation will be working on Windows operating system, Microsoft Visual Studio is suitable.

The SpadeShip Project employs a highly modular design. The modules and the sub modules form the infrastructure of the project and every module lies in the corresponding directory in CVS repository. Besides the code base, the documentation of the software, project data and website are placed as modules to facilitate change tracking.

## 3. THE CM PROCESS

### 3.1 Identification

The configuration items are separated into three different groups to provide an efficient management of the project. These groups are (1) *engines and modules*, (2) *data*, (3) *documents*.

#### 3.1.1 Engines and Modules

- Simulation Engine: core engine of the system
  - Server side simulation engine
  - Client side simulation engine
- Network Engine: establishes network communication
- AI Module: handles the movements of NPC



- Graphics Engine: handles all rendering operations
- Sound Module: plays the audio and voice messages
- Physics Module: contributes to the reality of the simulation

### **3.1.2 Data**

- Map
- Model
- Character models
- Character animations
- Texture
- Music

### **3.1.3 Documents**

- Design Report
- Configuration Management Plan
- Installation Manual
- User Manual

## **3.2 Management and Control**

### **3.2.1 Engineering**

During the weekly meeting, CCB holds formal technical reviews to inspect the suitability of the software for its intended use and identifies the deviations from specifications and standards. These FTRs enable the team to

- To verify the consistency between the design and the implementation
- Propose a roadmap to comprise in case of incapability
- Share their experience on their work, reflecting its effect on the overall work so that every member can get a sense of where the project is leading

### **3.2.2 Development**

In order to maintain the development management and control, Maça Yazılım employs

- Coding conventions to facilitate a standard which avoids miscommunication and eases the burden of reading other people's code.



- Code documentation following a common standard to ease the job of developer trying to learn the module.
- Document identification to inform another developer about the module being observed. The source code documents will have headers naming the coder, describing the module, and history information to easily skim over the change details.
- Naming conventions for modules so that confusion is avoided and necessary modules can be located easily.
- A standard IDE (Visual Studio) to facilitate the automation of version control, build system.

### **3.2.3 System Management**

Maça Yazılım team ensures system management and control by

- Utilizing CVS to facilitate version tracking and documentation and also allowing concurrent development on the project. CVS also allows the team to verify the implementation process.
- Holding weekly meetings to supervise the overall process. Every member can comment on the current status of the process and may brainstorm to improve its quality.
- Utilizing other means of communication (such as email, mail group) in a systematic way (with templates). System is free of subjective flaws.

### **3.2.4 Change Request**

All change requests of the team members will be evaluated to improve the project over the implementation process. Whenever one of the members has a change request, he informs the other team members of the request and reason for it thorough email. The request is added to the main topics which will be discussed in the next meeting of the team. Requests are made with a predefined template. The template consists of the:

- Name of the team member making the request
- Date of the request
- Modules along with their version numbers effected by the request
- The reason and description of the change request

*Change request evaluation:*





The confirmation of all members is necessary to accept or reject the request. This rule brings integrity and cohesiveness to the team. When the request is approved, an assignment bill is formed and emailed to every member. This bill consists of:

- Date of the assignment
- Name of the members assigned with the change
- List of the modules to be modified
- Description of the assignment
- The priority of modification
- Deadline for the modification

#### *Applying Change:*

If the change request is approved, the implementation task is distributed between the team members depending on their roles. The CI is checked out by the assigned members and necessary implementations are carried out to apply the change. Testing and auditing steps are performed, and then the new version is saved. For ease of following the different versions, version numbers are used (like v.1.1, v.1.2 ...). The version number consists of two parts: one signifying the major revision number and the other signifying minor revision number.

### **3.2.5 Build and Deployment**

The project is developed on Visual Studio environment and the solution files of Visual Studio are employed to provide automatic building functionality. The solution and project files are also located in the CVS server, so that building process can be improved or defects can be removed.

The project data, executables and documentation are bundled as self installing msi executables (Microsoft Installer) which are created on Visual Studio IDE. These packages are regularly deployed on the website to the audience of the software.

### **3.2.6 Defect Tracking**

All members of the team are responsible for testing and reporting the defects in the software. The team will maintain a history of defects in the mail group to facilitate the proper notification among team members. The defect report template is as follows:

- The name of the member reporting the defect
- The date of the report



- The symptoms of the defect
- The revision numbers of modules on which the defect is observed
- The priority of the defect
- A recipe for the reproduction of the observation of the defect

However, in case a defect tracking system is placed on department's machines, the team will switch to that system enhancing its capability of defect tracking.

### **3.3 Configuration Status Accounting**

As new versions are created, it is necessary to keep track of these different versions. Changes between the versions have to be recorded to prevent confusion in later stages. Therefore a method is determined to achieve this. A log file will be kept to record the modifications, and it will be updated on each change. Besides the logging facility of CVS server, members will be notified via email about the details of change. A template will be employed to avoid miscommunication among members. The template logs:

- The name of the team member committing the change
- The date of change
- The revision of the items and their names effected by change
- The description of change

These reports will facilitate communication among team members and will keep everyone up to date with the current status of the project.

The release will have change logs which will contain information about the updates characterized by the release.

### **3.4 Auditing**

In auditing, the project will be evaluated and actions will be taken depending on the evaluation results accordingly.

For the project, the auditing will be done after all releases and important changes. In addition to these usual audits, members can request for audits in the meetings.

There will be mainly three types of auditing: functional audits, physical audits and process audits. Functional audits will be done to test the software to see if it performs in accordance with requirements in the baseline. Physical audits will be done to have assurance about that the



software contains all of the required components, documents and data. The process audit will compare and contrast the manner in which the end product is produced to the written procedures. It will also consider the validity and overall reliability of the process itself. The auditing reviews will be logged and emailed to every team member. The content of the document depends on the type of auditing performed.

#### **4. PROJECT SCHEDULES – CM MILESTONES**

During the development process, team members will need to update CMP as the project will not remain in the same status. There will be three releases and the team is planning to update CMP after each release.

The CM milestones are given in the table below:

Date	CM Milestone
09.03.2008	Delivery of CM
27.03.2008	Update of CM
24.04.2008	Update of CM
13.06.2008	Final CM

#### **5. PROJECT RESOURCES**

The following items will be used as project resource:

- TortoiseCVS Client
- Jalindi Igloo Visual Studio CVS Plug-in
- SSH Protocol (Secure Shell Client)
- Web Server
- Mail group

for CM activities.



## **6. PLAN OPTIMIZATION**

In order to have a frame while working for development of the project, it is needed to keep the CMP valid. For this reason CMP will be updated whenever required. Since we have meetings very frequently (at least once a week), all the members are informed about the changes in a short period. During the CMP updates, a plan optimization will also be performed.

