Configuration Management Report
for
Virtual Turkey Project

MECAC
Cinar Kilcioglu
Mert Degirmenci
Umit Cavus Buyuksahin
# TABLE OF CONTENTS

1. Introduction  
   1.1 Purpose of Configuration Management Plan  
   1.2 Scope of Document  
   1.3 Definitions, Acronyms and Abbreviations  
   1.4 Document References  
   1.5. Overview  
2. The Organizations CM Framework  
   2.1. Organization  
   2.2. Responsibilities  
   2.3. Tools and Infrastructure  
3. Configuration Management Process  
   3.1. Identification  
      3.1.1 Source Code  
      3.1.2. Data  
      3.1.3. Documentation  
   3.2. Configuration Management and Control  
      3.2.1 System Change Request  
      3.2.2 System Change Evaluation  
      3.2.3 System Change Implementation  
   3.3. Accounting  
   3.4. Auditing  
4. Project Schedules and CM Milestones  
5. Project Resources  
6. Plan Optimization
1. Introduction

1.1 Purpose of Configuration Management Plan

Software Configuration Management (SCM) is a formal engineering discipline that, as part of overall system configuration management, provides the methods and tools to identify and control the software throughout its development and use. SCM activities include the identification and establishment of baselines; the review, approval and control of changes; the tracking and reporting of such changes; the audits and revive of the evolving software product and the control of interface documentation and project supplier SCM.

The project lies in a dynamic structure. As the project progresses, new specific needs arise. In order to meet these needs, the project always has to be agile and easily tractable. By considering all of these aspects, for Virtual Turkey project, which has numerous components changing in time, configuration management is vital. SCM would help a strong communication between the developers and this facilitates a healthy project development.

In short, the purpose of this document is to keep maintainability of the project during the development cycle.

1.2 Scope of Document

The scope of this document is about the identification of configuration management plan for project ‘Virtual Turkey’. CMP presents configuration management activities that will be applied throughout the development process of our project and these activities explained in this document are applicable during the development and maintenance phases of the project.

The document contains methodologies and activities designed to manage changes and relate them. Since the project includes a medium number of modules, includes some integrated libraries and is being developed by a group of five people. The related audience for this document is MECAC software team, our assistant and instructors.

1.3 Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>CCB</th>
<th>Configuration Control Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI</td>
<td>Configuration Item</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
</tbody>
</table>
1.4 Document References


1.5. Overview

Section 2 describes the organizations of CM framework. General organizational context, responsibilities of the group members and the tools and infrastructures needed are explained. Section 3 mainly explains the methodologies of CM processes. The rest of the report, project schedule and CM milestones, resources needed, and optimization of CM are stated respectively.

2. The Organizations CM Framework

2.1. Organization

The organizational units are formed with each group member's fully devotion. The units serving to SCM activities and their descriptions are as below:

Configuration Manager: The configuration manager is responsible for supervising the efforts of the project related to CM.

Development Team: Development team is responsible for developing the network and graphics parts of the project. It is their responsibility to meet the software requirements specifications according to software development plan.
Testing Team: Testing team is responsible for testing whether the components meet the predefined requirements, and informing the development team about the deficiencies of the project, which come to light in testing phase.

2.2. Responsibilities

The organizational units described in Section 2.1. is responsible for maintaining SCM in all phases of the project. Communication between the teams is indispensable for a well-developed project.

2.3. Tools and Infrastructure

- Windows NT Operating System Family: Windows NT is a family of operating systems produced by Microsoft. The recent members of this family are Windows XP, Windows Vista, and 7. The main infrastructure of the project is a Windows NT operating system.

- Microsoft .NET Framework: The .NET is a software framework for Microsoft Windows operating systems. It includes a large library, and it supports several programming languages. The project code is mainly written in C# programming language. CM will require this programming language as the improvements will be made in C#, which requires .NET framework.

- SVN: Apache Subversion is a software versioning and a revision control system. Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System(CVS). SVN is the vital part for a CM system. It provides maintenance of the code's being secured.

- Trac: Trac is an open source, web-based project management and bug-tracking tool. It has the ability to interface with Subversion. Trac provides the development of the project’s being traced.

3. Configuration Management Process

3.1. Identification

Configuration items are crucial to identify the project. These items can be categorized into three categories: Source Code, Data and Documents.
3.1.1 Source Code

The most important part of the project is coding phase that all team members make contribute to. Each CI is assigned a short mnemonic identifier according to its functionalities. The components of the project corresponds the CI of the project that are expressed below:

- Client Network Component: This component of the project describes how the clients communicates with the server through the network.

- Server Network Component: This component is responsible for updating the information of clients while balancing the network load on the server.

- Physics Component: This component is responsible for physics calculation for persistent world of the game.

- Management Component: The management component will provide the utilities that system administrators need. It encapsulates several crucial processing units of the game such as data management, non-playing characters, and game loop.

3.1.2. Data

Data of the project are mainly Character and NPC data objects. Account, Vehicle, Treasure data objects are associated with Character object. NPC object, on the other hand, stores the information about the trade items of it. These data objects are the essential CI. Picture files, music files are other main type of data of the project. Although no major change is planned to be done on them, for group members, all data will be available on the server during the development of the project, which provides each member to test the project anywhere.

3.1.3. Documentation

For Virtual Turkey project, documentation is a mandatory CI. Since the project will be open source in near future, documentation is perhaps the most important part in development cycles. Following documents have been created up to now.
- Reports:
  - Project Proposal
  - Requirements Analysis Report
  - Initial Design Report
  - Detailed Design Report
  - Configuration Management Plan

- Web Documents:
  - Web page
  - Web blog

- Development Process:
  - Weekly reports

- Users Manuals
- Tutorials of End Product

### 3.2. Configuration Management and Control

#### 3.2.1 System Change Request
SCRs can be bisected as minor SCRs and major SCRs. Major SCRs that Trac System controls consist of name of member, date, description, deadline, priority, version, special notes. Minor SCRs are included to the project directly, and items are checked into SVN.

#### 3.2.2 System Change Evaluation
Minor SCRs is maintained on the project by the members of the project without assessment. However, evaluation of Major SCRs are discussed both face-to-face in team meetings and by using tickets on Trac System.

#### 3.2.3 System Change Implementation
After concluding an evaluation about SRCs, all CIs that have to be changed and updated are determined, and these changes are applied to related CIs by committing related resources via SVN.

### 3.3. Accounting
To record configuration data and to present a coherent picture of the product at determined moments in the product’s life-cycle, to establish and maintain configuration records for CIs, configuration status accounting has been done in Virtual Turkey project.
All the developments made in the project will be done on SVN, and be available to trace. To have a better understanding of the changes made before, applied improvements will be documented in weekly reports and project website. This will enable the group members to not only see the final version of the project, but also comprehend what other group members have been doing on the project. Moreover, it will provide the assistant of the project group to evaluate the performance of the group.

3.4. Auditing

Mert has the responsibility for maintenance of the reliable configuration version. Important changes are notified to team members by Mert via meetings or other suitable channels such as Internet, phone call etc. The group reviews the state of the project and makes required modifications to the current configuration.

4. Project Schedules and CM Milestones

We have divided the development in components to ease workload sharing. The milestones of Virtual Turkey are the followings:

- Delivery of CM: 20 March 2011
- First Development Snapshot, Demo: 05 April 2011
- First Release, Demo: 15 May 2011
- Final Release, Demo: 28 May 2011

5. Project Resources

Main resources have been used and going to be used for an efficient SCM are as follows:

- Lidgren
- XNA
- SVN
- TRAC
- Project Website
6. Plan Optimization

We divided our CM responsibilities between our three members according to their interests. Each member is responsible for himself. We mostly communicate via phone call. In addition, we are making regular weekly meetings for more control over the group and also we have weekly meetings with our assistant. Moreover, we control our process with small demos for consistency of our project.

CMP will be a guide for coordination and progress of Virtual Turkey. Any updates or changes in CM schedule will be controlled by CCB. When an update or change occurs, this will be followed by all group members via TRAC. There will be meetings regularly to keep track of progress of configuring items. According to these meetings, the living schedule will be updated regularly. These meetings will keep us all up to stick with the schedule but as in all software development processes, there can and will be some unpredicted problems.