

MIDDLE EAST TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CENG 492
SENIOR PROJECT IMPLEMENTATION

Spring 2005

Software Configuration Management Plan



WIRTUAL WISION
'Beyond The Sight'

Dođan Sever
Ercan Őahan
Alper Taő
Kubilay Timuđin
Rakıp Yaőar

March 2005

Table of Contents

1. INTRODUCTION	1
1.1. Purpose of Software Configuration Management Plan	1
1.2. Scope of the Document	1
1.3. Terms, Acronyms, Abbreviations	1
1.4. Document References	1
2. ORGANIZATION'S CONFIGURATION MANAGEMENT FRAMEWORK	2
2.1. Organization.	2
2.2. Responsibilities	2
2.3. Tools & Infrastructure	3
3. CONFIGURATION MANAGEMENT PROCESS	3
3.1. Identification	3
3.2. Management and Control	3
3.3. Configuration Status Accounting	4
3.4. Auditing	5
4. PROJECT SCHEDULE	5
5. PROJECT RESOURCES	5
6. PLAN OPTIMIZATION	5
7. APPENDIX PART	6

1. INTRODUCTION

1.1. Purpose of Software Configuration Management Plan

Software Configuration Management (SCM) is a vital part within the progress of a software development project. This is because change is inevitable in the development cycle. Any change in development plans has to be dealt with carefully so that the progress stays under control. SCM plan helps us identify, manage, control and audit the changes correctly. The changes have to be implemented correctly and every member involved in the project has to be informed about the changes. Otherwise, it might inevitably result in an unsolvable confusion. This is why a consistent SCM plan is needed. Regarding these facts, this SCM plan is going to be applied in order to identify changes, control changes, assure the correctness of the implementation of identified changes and lastly report the changes to everybody.

1.2. Scope of the Document

The scope of this document is the identification of Software Configuration Items (SCI), management of change control, auditing the changes and reporting the changes in order to inform the involved people in the project. By this way, the changes in the project will be made without any confliction. Next section covers the organizational units which participate in all SCM activity, roles of these organizational units and relationships among these organizational units. The third section covers configuration management processes: identification of configuration items, configuration management and control, configuration status accounting and configuration auditing. Section four refers to the project schedule. Fifth section is about project resources which are tools, techniques, equipment needed to implement the SCM activities specified in the SCM plan. Section six covers plan optimization.

1.3. Terms, Acronyms, Abbreviations

CCB,	Configuration Control Board
CI,	Configuration Identification
CM,	Configuration Management
CSA,	Configuration Status Accounting
CVS,	Concurrent Versions System
ERB,	Engineering Review Board
PCA,	Physical Configuration Audit
SCI,	Software Configuration Item
SCM,	Software Configuration Management

1.4. Document References

IEEE Std 828-1998 Software Configuration Management Plan Standard and the Software Configuration Management document covered in the class are followed while developing this document.

2. ORGANIZATION'S CONFIGURATION MANAGEMENT FRAMEWORK

In this part the organizational structure of Wirtual Wison – 3D Adventure Game is explained. The subtitles covered here are:

- Organization
- Responsibilities
- Tools & Infrastructure

2.1. Organization

The organizational units related to SCM activities in the project are as follows:

- Configuration Control Board (CCB)
- Configuration Management (CM) Group
- Engineering Review Board (ERB)
- Testing Team
- Developer Team

All five members of the project are involved together in each of the above groups. The role of each organizational unit is as follows:

CCB: Reviews, denies or approves all the changes. Follows all the changes and ensures that they are done correctly. Approves all CI baselines.

CM Group: Creates and maintains the CMP. Coordinates and reports all the CM activities. Coordinates the development team for the changes.

ERB: Does the technical analysis and evaluation of the change plans. Evaluates the change proposals in the base of cost, time and performance and gives recommendation to the CCB.

Testing Team: Makes the change requests after the results of the testing process.

Developer Team: Creates the baselines and releases. Implements all the CM activities which are the part of the actual system development.

CCB is on top of the organizational unites' hierarchy. All other units are supervised by CCB. The activities of the CM are followed by CCB and their requests are either approves or denied. CM also coordinates the developer team. ERB evaluates the change requests and gives recommendation to the CCB. Developer team implements all the change requests that come from CM.

2.2. Responsibilities

The SCM activities and the corresponding organizational units are as follows:

SCM Activities	CCB	CM	ERB	Testing Team	Developer Team
Configuration Identification	Review	Originate			
Configuration Control	Review, Approve		Originate, Review	Test	Originate, Implement

Configuration Status Accounting	Review, Approve	Originate	Review		
Configuration Auditing	Originate, Review	Originate			

Table 2.1 - SCM activities

2.3. Tools & Infrastructure

CVS will be the project's version control system. By using CVS, the history of the source files will be kept. With CVS, the configurations to the source code and the documentation will be done in a sequence and the crashes will be avoided.

3. CONFIGURATION MANAGEMENT PLAN

In this section, some details about the SCM tasks that are going to be achieved in development cycle are given. Configuration Management activities include all tasks required in managing the Configuration Items of the system. These tasks can be listed as:

- Configuration Identification
- Configuration Management and Control
- Configuration Status Accounting
- Configuration Auditing

Change Control Process Flowchart of CM Process can be found in Appendix Part.

3.1. Identification

In this subsection, physical and functional characteristics of the configuration items are identified, named and described. There are four configuration items in the project:

- Hardware: involves all the physical equipments which are required through the development of the project.
- Software : involves all non-document files required for project development
- Baselines: involve the milestones in the SCM. CVS will be used for controlling changes.
- Documents: involve all the documents developed through the project development.

3.2. Management and Control

In this subsection, the process of requesting, evaluating, approving or disapproving proposed changes to a baseline are described. The steps which are followed in configuration control are as follows.

Change Request: Since the group is composed of five people, the change request comes from one of the group members. Then members come together and discuss whether the requested change is necessary or not.

Approving or Disapproving Requested Changes: If the change is found unnecessary, the request is rejected and all group members are informed about it. On the other hand, if the change is found reasonable, the request is accepted.

Implementing Changes: If the change request is accepted, the implementation phase of changes starts. First, the baseline configuration item from project database is checked out and changes are applied. Next, the updated item is tested for quality assurance. Then the configuration auditing is performed and finally the new version of the software is checked in to the project database and distributed to every group member.

Version Control: After the changes are made, the version number of the software should be adjusted. Universal version number system is used. Our version numbering system will be as:

<major change>.<minor change>

When a minor change happens we will only increase the minor change field the version number. In major changes the major change field will be increased.

Version control will be used for the entire length of the project and will be implemented via CVS. All files for all phases will be stored in CVS. CVS' basic version control functionality maintains a history of all changes made to each directory tree it manages, operating on entire directory trees, not just single files.

Firstly, each developer uses CVS checkout to create his own copy of the source tree from the CVS repository. The command can operate on a directory tree, a single file or a module; a module groups several files or directories into one entity which can be operated on as a unit.

The developer changes, compiles and tests the code in his copy of the source tree (called a working directory) with whatever tools he chooses. Then the developers can use CVS add and CVS remove to add and remove files.

When the changes are complete, the developer uses CVS commit to merge his changes back into the repository which will make his changes available to other developers.

At any point, the developer may use CVS update to merge changes committed by others into his working directory. If there are uncommitted changes to files in his working directory, CVS prints a message and attempts to merge the changes from the repository with his changes in the working directory. If the merge fails, CVS indicates a conflict which it resolves manually with a text editor.

3.3. Configuration Status Accounting

Configuration Status Accounting (CSA) is the process for recording, monitoring, and reporting information necessary to manage all configuration items associated with a specific baseline throughout their life cycle. CSA consists of creating reports and of keeping a history of CM activities. For every change the following information is made and recorded:

- Detailed description of the change: The change made will be explained. Why this change is needed and how it is made will be recorded.
- Analysis of the change: After the change, what changed in the system and how the system works after the change.
- Names of people making the change: The names of the group members who made the changes.
- Title, type, release and version, with date: The title, type of the changes and the release and version numbers of the system after the changes.

3.4. Configuration Auditing

In this section, what extent the actual SCI reflects the required physical and functional characteristics are determined. Whether the specified changes have been made correctly, whether a formal technical review has been conducted to evaluate technical correctness of the changes, whether the software engineering standards have been applied correctly and whether we have recorded information about the changes correctly are determined. Functional, physical and process audits will be done. Functional audits will be done before acceptance of each functional baseline and after system testing and before delivery of the release. The Physical Configuration Audit (PCA) will be held prior to any system release. And during the change process at each step, process audits will be held.

4. PROJECT SCHEDULE

CM milestones will be added in the project schedule. After implementing each module, all the CM activities will be made. In the schedule, milestones are shown as different modules. During testing phase, CM activities will be performed for each change. Project schedule is given in the Appendix Part.

5. PROJECT RESOURCES

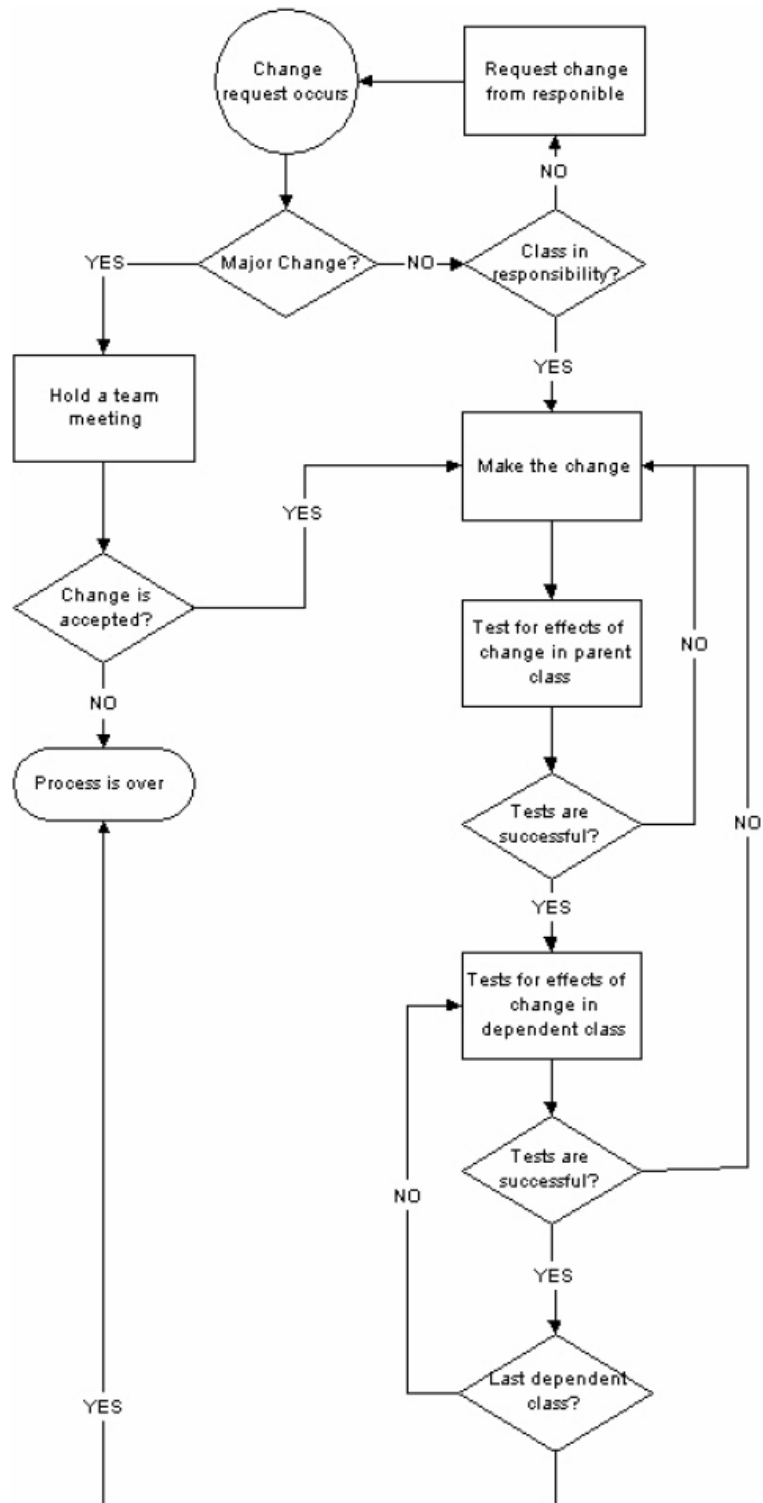
CVS will be used as CM tool. The group members will get the latest version of the project by using check out feature of CVS. After the change has been finished, the latest version will be updated to CVS repository so that each group member will have access to the latest version of the project. Since CVS does not lock the resources for updating, members can work simultaneously on the same files. In case of conflicts, CVS warns the developers so that conflicts are avoided. It also provides extensive logging and auditing facilities.

6. PLAN OPTIMIZATION

Since this is a small group composed of five people, everybody in the group will be responsible in maintaining the SCM plan. The plan will be updated when needed in order to maintain CM activities effectively. Once the updates are done to the plan, CCB will control the updates.

7. APPENDIX PART

Change Control Process Flowchart:



Schedule

ID	Task Name	Duration	Start	Finish	Resource Names
1	Team Construction	7 days	Mon 04.10.04	Sun 10.10.04	Management;Reporters
4	Scope	5 days	Mon 11.10.04	Fri 15.10.04	Management;Project Manager
7	Research and Survey	6 days	Sat 16.10.04	Thu 21.10.04	Researchers;Reporters
11	Requirement Analysis	15 days	Fri 22.10.04	Fri 05.11.04	Researchers;Analysts;Reporters
17	Development Tools	5 days	Sat 06.11.04	Wed 10.11.04	Researchers;Project Manager;Developer
21	Game Design	21 days	Thu 11.11.04	Wed 01.12.04	Management;Developers;Analyst;Art Dir
31	Game Design Tuning	12 days	Thu 02.12.04	Mon 13.12.04	Revisors;Developers
35	Project Presentation	5 days	Mon 13.12.04	Fri 17.12.04	Presentation Team
36	Final (Detailed) Level Design	23 days	Sat 18.12.04	Sun 09.01.05	Developers;Reporters
39	Game Design Complete	1 day	Mon 10.01.05	Mon 10.01.05	Developers;Project Manager
40	Prototype	11 days	Sun 09.01.05	Wed 19.01.05	Developers;Presentation Team
43					
44	Multimedia Implementation	10 days	Wed 16.02.05	Fri 25.02.05	
45	Create Graphics	10 days	Wed 16.02.05	Fri 25.02.05	All team
46	Cretate Audio	10 days	Wed 16.02.05	Fri 25.02.05	All team
47	Implementation	47 days	Sat 26.02.05	Wed 13.04.05	
48	Basic Classes for Models	6 days	Sat 26.02.05	Thu 03.03.05	Doğan
49	Game Engine Implementation	6 days	Thu 03.03.05	Tue 08.03.05	Rakıp
50	Graphics Engine and Sound Classes Impl.	6 days	Tue 08.03.05	Sun 13.03.05	Ercan
51	First Development Snapshot	1 day	Mon 14.03.05	Mon 14.03.05	All team
52	Menu and Configuration Classes Impl.	5 days	Tue 15.03.05	Sat 19.03.05	Alper
53	AI Engine Implementation	5 days	Sun 20.03.05	Thu 24.03.05	Kubilay
54	Integration	20 days	Fri 25.03.05	Wed 13.04.05	All team
55	Map Builder Implementation	10 days	Thu 14.04.05	Sat 23.04.05	
56	GUI Implementation	5 days	Thu 14.04.05	Mon 18.04.05	Rakıp
57	Data Abstraction and Export Layers Impl.	5 days	Tue 19.04.05	Sat 23.04.05	Ercan
58	Integration of all Codes and Tuning	9 days	Sun 24.04.05	Mon 02.05.05	All team
59	First Release	1 day	Mon 02.05.05	Mon 02.05.05	All team
60	Testing	17 days	Tue 03.05.05	Thu 19.05.05	
61	Game Engine Testing	5 days	Tue 03.05.05	Sat 07.05.05	Rakıp
62	Initial Game Testing	12 days	Sun 08.05.05	Thu 19.05.05	
63	User Interface Testing	2 days	Sun 08.05.05	Mon 09.05.05	Doğan;Alper
64	Audio Testing	2 days	Thu 12.05.05	Fri 13.05.05	Kubilay
65	Game Play Testing	6 days	Sat 14.05.05	Thu 19.05.05	All team
66	Project Presentation	1 day	Fri 20.05.05	Fri 20.05.05	Ercan;Doğan;Kubilay
67	Final Testing	7 days	Sun 22.05.05	Sat 28.05.05	All team
68	Final Release	8 days	Sun 29.05.05	Sun 05.06.05	All team
69	Documentation	2 days	Mon 06.06.05	Tue 07.06.05	
70	User Manual	2 days	Mon 06.06.05	Tue 07.06.05	Ercan;Alper;Kubilay
71	Programmers' Documentation	2 days	Mon 06.06.05	Tue 07.06.05	Ercan;Alper;Kubilay
72	Final Demonstration	1 day	Fri 10.06.05	Fri 10.06.05	All team

END OF DOCUMENT