CENG 490

STARSOFT

Configuration Management Plan

By;

Mehmet ALBAYRAK
Ömer ESER
Özer GÜMÜŞ
Özge YAMASAN
Fatih YILDIRIM

1. INTRODUCTION

1.1. Purpose of the Configuration Management Plan

Life is changing like our software projects. Almost every part of the life, one can come across with a change in job during the development stage, as well. It is really an important problem to adopt software to the highly developing technology. In the global world today, many applications have been changing and adding some new parts by the developer firm of that software. It is the situation, then what about the development stage! It is certain that a software developer or team will change not the general points but some internal points due to some emerging requirement maybe caused by the technology or customer need. If that is the situation, developer or team members must apply some predefined rules to make these changes applicable to the project. Configuration Management (CM) forces developers to make changes according to some rules and that is the purpose of CM plan document. Thanks to the document, we will more control on the project when applying some changes and reporting them.

Moreover, thanks to this document, team members will have some more roles on project development that every member will be more careful when making changes and reporting them and some more work like backing up will be required, as well.

1.2. Scope of the Document

CM as noted above is a must for long term projects to apply changes during project development time. It is an important rules concept for our project “Ajax Developer Studio” that will be prepared by Starsoft Company. There are some important reasons that make CM be essential for us. The first one is that we have been developing as stated above a developer studio with many components and that is a time required job. It will make easy to change some parts formally. The next reason is that our project has many modules that are connected and communicate with each other in a suitable way. CM will be useful at that part of the project to combine each module with others and work properly.

1.3. Definitions, Acronyms and Abbreviations

There are some definitions, acronyms and abbreviations used in CM. These are:

Acronyms	**Definitions**
CCET	Configuration Control and Evaluation Team
CMT	Configuration Management Team
CM	Configuration Management
CMP	Configuration Management Plan
TT	Testing Team
DPT	Debug Process Team
DT	Developer Team
SCR	System Change Request
CVS	Concurrent Versioning System
CI	Configuration items
FCA	Functional Configuration Audits

1.4. Document References

1. IEEE Standard for Software Configuration Management Plans, IEEE Std 828 – 1998
2. IEEE Standard Glossary of Software Engineering Terminology,IEEE Std 61012-1990
3. IEEE Standard for Software Quality Assurance Plans, IEEE Std 730-1998

1.5. Document Overview

This is a CM that is constituted from some subtitles:

- In the first part that is introduction, purpose and scope of the document (CMP) are explained. Reason for the changes during development stage and required rules to apply these changes in a formal way is explained in a clear way. Some acronyms, definitions and abbreviations are given, as well.

- In the second part of CMP, the organizational structure is explained in a detailed way. In this part of CM, responsibilities are given to members of project team and it is also explained. How team members will communicate each other is also described in this part.

- In the third part of CMP, configuration management process is covered. This section includes the subsections of identification, configuration management and control, version control, configuration status accounting and configuration auditing. The rules decided about the processes are documented.

- In the next part of CMP, project schedule is clearly explained.

- In the next part will give information about the project resources.

- In the final part of the CMP will give more specifics on project plan in a detailed way.

2. THE CM FRAMEWORK

2.1. Organization

In our project, CM activities are handled by the following teams:

- Change Evaluation Team (CET)
- Configuration Management Team (CMT)
- Testing Team (TT)
- Development Team (DT)

In big companies, the members of these teams are determined and they are effectively used, but since we have only five people we can not make such a complete organization. So, every member of our project team will be a member of each team.

2.2. Responsibilities

Team responsibilities are as follows:

- Change Evaluation Team (CET): This team is responsible for the evalution of SCRs. According to time and effort costs, CET either accepts or rejects the SCR.

- Configuration Management Team (CMT): This team is responsible for updating the CM schedule according to performed activities. Team is also responsible for making the team members obey the CM schedule.

- Testing Team (TT): This team is responsible for proper functioning of the software. Team can make SCRs if it is necessary.

- Development Team (DT): This team is responsible for implementing the source code of the software.

2.3. Tools and Infrastructure

Our project team will use the following tools:

- CVS
- NetBeans

2.3.1 Concurrent Versioning System (CVS)

CVS is an open source version control system. It has a central repository that includes current source code, past versions of the system, and logs that document changes to the system. Every member of the project team will be able to reach that repository from his own computer.

2.3.2 NetBeans

NetBeans, basically, is an integrated development environment for Java. It has a clean interface and offers a large number of opportunities like code completion assistance, folding source code portions, and text highlighting.

3. THE CONFIGURATION MANAGEMENT PROCESS (CMP)

The following phases are followed in CMP.

- Configuration Identification
- Configuration Management and Control
- Version Control
- Configuration Status Accounting
- Configuration Authentication

3.1. Configuration Identification

The components that make up the software with their documentation and data compose the Configuration Item shortly called the CI. Keeping in mind that the status of the CI's at a given time are called baselines, they are given special consideration. The baselines of our project are planned to have the following attributes (in accordance with Traditional Software Configuration Theory)

1. Functionally complete

   The functionality of a baseline will be well-defined. The features and functions of a particular baseline will be documented be kept available for further reference. Therefore the properties and capabilities of a baseline will be clearly known.

2. Known Quality

   The quality of a baseline will also be well defined. For example bugs will be documented and further testing will be implemented before establishing a baseline.

3. Immutable and completely recreatable

   Once a baseline is defined it is going to be immutable in the sense that it can not be changed. There is not much flexibility at the point. Version Control will also be put to use so that the baseline can be recreated at any point.

This phase lists the categorization of CI's. The CI's that will be used in our project are as follows.

- Documents
- Baselines
- Hardware
- Software

3.1.1. Documents

After formal approval process , all documents will be placed into the document library and the official website.

3.1.2. Baselines

In accordance with CVS, baselines will be will be tracked, audited, retained and version controlled.

3.1.3. Hardware

Hardware is the physical environment for implementing, testing, designing and running the software. Hardware CI's are listed and documented. These documents will be reported as necessary.

3.1.4. Software

Software CI's include the files to implement the system components of the software. CIs are documented and these documents are placed in the related reports.

3.2. Configuration Management and Control

Our Configuration Management Process consist of ;

- Change Requests
- Evaluation of Change Requests
- Implementation of Change Requests

3.2.1. Change Requests

The requests are to be kept in the CM_Change_Requests folder at CVS and taken care of during meetings. There are folders used for resolved and unresolved requests, as well. Change requests can only be opened for a previously completed module and must be in accordance with Request document template.

3.2.2. Evaluation of Change Requests

The evaluation process is done weekly during the meetings. The changes will be managed and handled as they will be put the the related folder(Resolved\Unresolved).

3.2.3. Implementation of Change Requests

Once the request is approved, the related action will be put to use by the owner of the module and configuration item version will be changed accordingly in CVS. After the changes testing will be implemented.

3.3. Version Control

CVS, which is provided by our department, will be used to keep track of the changes on the source files. It will keep our configuration items in its repository. There will also be local copies. These copies will be committed to CVS regularly. The following procedures will take place.

1. Check out a file
2. Update the file
3. Commit the file into the repository of CVS
4. CVS changes the version automatically

3.4. Configuration Status Accounting

This process is the bookkeeping process of each release. Tracking what is in each version of software and the changes that lead to this version are done in this procedure.

In our project documentation of these processes will be done and this process will be easy to follow.

3.5. Configuration Authentication

The configuration authentication is an audit performed before it is opened to the users. Verification of functional aspects, documentation and data can be completed in this phase.

- Functional Configuration Audits (FCA)
- Peer Reviews (PR)

3.5.1. Functional Configuration Audits (FCA)

FCA will be used to determine if the implementations of each CI item are in accordance with its design specifications. Also the verification of the consistency between proposed functional baselines and the changes is done.

3.5.2. Peer Reviews (PR)

Having one or more programmers review the source code of a program written by someone else is the Peer Review part of our project. Better code is produced, since others have already tried to understand it.

We will use our weekly meetings with Oral Dalay as our Peer Review opportunity. The feedback we receive in the meetings are going to be discussed and put to use.

4. PROJECT SCHEDULES & CM MILESTONES

CM Schedule gives information about the sequence of tasks and coordination of the CM activities. Our living schedule, made in beginning of the semester, consist of task of the project. This schedule is on the website and it is updated regularly.

Team members will meet with AJAX supervisor on every Thursday. During this meeting, we will discuss current state of our project. In addition, after this meeting our members will come together to discuss the current situation of development cycle and to ensure everything going as it should.

CM milestones will be added in the project schedule. During implementing each module, all the CM activities will be made. During testing phase, CM activities will be performed each change. These milestones are the followings with the deadlines.

• Debugger module implementation	- 04.04.2007
• Editor module implementation	- 07.05.2007
• Project module implementation	- 07.05.2007
• Database module implementation	- 08.05.2007
• Integrations of modules	- 08.05.2007

- Testing - 04.06.2007
- Documentation support - 11.06.2007

5. PROJECT RESOURCES

The backup of the project will be achieved by using CVS system. The system will be used carefully in that some files due to their big content and that don't need much change (like the resources files). All the changes to the files in CVS system will be made according to the decisions of the project members. Such decisions will be handled in the meetings or through the shared e-mail.

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

Our configuration management plan will be the main road map for our development process. We are all responsible for monitoring and maintaining CMP. Our progress will be stringly compatible with our plan. We will do critical reviews of CMP at every milestone and at the end of every big module of the project. These reviews prevent the CMP becoming redundant and outdated. Furthermore, the mail traffic among the members guides them about the changes which is considered as important and will done during the development.