



AJAX DEVELOPMENT ENVIRONMENT

CONFIGURATION MANAGEMENT PLAN



1. INTRODUCTION.....	3
1.1 PURPOSE OF THE DOCUMENT.....	3
1.2 TEAM AND PROJECT OVERVIEW.....	3
1.3 SCOPE OF THE DOCUMENT.....	4
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS	4
1.5 DOCUMENT REFERENCES	4
2. THE ORGANIZATIONS AND CM FRAMEWORK.....	5
2.1 ORGANIZATION	5
2.2 RESPONSIBILITIES	5
2.3 TOOLS & INFRASTRUCTURE.....	5
3 THE CM PROCESS.....	6
3.1 IDENTIFICATION	6
3.1.1 Software	6
3.1.2 Baselines.....	7
3.1.3 Documents.....	7
3.2 MANAGEMENT AND CONTROL	7
3.2.1 Change Request.....	7
3.2.2 Meeting.....	8
3.2.3 Change Request Evaluation	8
3.2.4 Defect Tracking.....	9
3.2.5 Version Control	9
3.3 CONFIGURATION STATUS ACCOUNTING	10
3.4 CONFIGURATION AUDITING & REVIEWS.....	10
4. PROJECT SCHEDULES	10
5. PROJECT RESOURCES	11
6. PLAN OPTIMIZATION	11

1. INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

Purpose of this document is to demonstrate the planned details of the configuration management which will be applied during the development of the KAJAX IDE. All parts of this document are prepared by the group members and it will guide us to manage our pieces of products. This will prevent any conflicts between the team members' own ideas and hopefully saves us time.

1.2 TEAM AND PROJECT OVERVIEW

Our system is composed of 6 modules below.

- Editor Module
- Debugger Module
- Panel Module : Solution Explorer, File Explorer, AJAX Actions, Toolbox
- Database Module
- FTP Module
- CVS Module

All the modules in our project takes the input from a module and gives its output to a module, that is, all the modules are heavily related with each other. Because of the nature of the project and complexity of the modules when we want to make some modifications on the project and on its main functions it is really an important case to do these modifications under a controllable plan. So configuration management plan is not only a necessity, but it is a serious must to keep track of and control the interaction between the current status and the future modifications on the project.

Karinca Teknoloji team is a small team of four people, who are able to communicate easily. Group development tasks are obvious and these tasks are assigned to prevent cohesion between the developers.

For the above reasons our aim is to prepare a CM plan which is easy to understand and free of unnecessary details which consumes much more time than it saves. So it will be a simple set of rules and procedures.

1.3 SCOPE OF THE DOCUMENT

This documentation is for our software's CM plan. This includes the whole process of configuration management. In fact it provides the necessary reactions against modifications. When a change required, it is important to state the member of the group who will do it, the way of doing the change, the time of the change and the effects of it to other modules of the project. It is also essential to state and foresee any troubles after doing the change and guarantee not to occur by taking some precautions. So we will identify and give solutions to these problems in the following parts of this document.

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

The technical terms used in this document are defined in the following table:

ACRONYM	DEFINITION
CM	Configuration Management
CVS	Concurrent Versions System
CI	Configuration Item

1.5 DOCUMENT REFERENCES

- Software Configuration Management slides in CENG 492 course.
- Configuration Management Plans, IEEE Std 828-1998.
- Analysis, Initial and Detailed Design Reports by Karinca Teknoloji, Fall 2007, <http://senior.ceng.metu.edu.tr/2007/karinca>

2. THE ORGANIZATIONS AND CM FRAMEWORK

2.1 ORGANIZATION

At the beginning of the implementation process we wanted to assign the similar modules in terms of implementation to each member of our group. Volkan will be responsible for the editor module, Halil will do the explorers(solution and file) and the toolbox module, Caner will complete FTP and CVS module and Okan will do the database module and AJAX Action Panel. Since all the modules are done by different members and these modules are heavily related with each other, each team member will play an active role in configuration management. The assigned configuration manager is Ali Caner PARMAKSIZ. He is responsible for related to any procedure of configuration management. Ali Caner is also responsible for the communication among team members. Each team member will be informed of any change request and a change request will be taken into consideration by the group.

2.2 RESPONSIBILITIES

Any team member having a change request will inform Ali Caner. If the change is not an important one Caner will mail the other members this change and ask for their approval. Otherwise he will arrange a meeting and during this meeting the change request will be discussed. Any team member working on a process about configuration management will report the status of his job to Caner.

2.3 TOOLS & INFRASTRUCTURE

CVS server will be used for handling changes correcting defects and version control. Whenever a change is done, this change is put into CVS server in the control of Caner. If a code segment in the CVS server is subjected to change, this change approval will be decided by group consensus that is turning back to a history state which is before the current CVS server state is forbidden. MSN and mail will be used for communication.

We prefer to use 2 main applications as CVS clients since using pure CVS is unnecessarily boring. Every group member will access CVS server from his/her own computer by the help of these tools. The first one is TortoiseCVS. It allows us to work with files under CVS directly from Windows Explorer and it is freely available under the GPL. With TortoiseCVS, we can directly check out modules, update, commit and see differences by right clicking on files and folders within Explorer.

Our second CVS client is in fact a fully integrated development environment for Java, namely Netbeans. We are developing our product with Java and Netbeans is a very preferable CVS tool which works under both Linux and Windows.

3 THE CM PROCESS

3.1 IDENTIFICATION

Identification of CIs is important since the success of the production relies heavily upon configuration management activities. We have decided to use the following types of configuration items.

3.1.1 Software

Software is the main part of the project and consists of six modules. These modules are Editor, Debugger, Database, Panel, FTP, CVS modules. Each module will contain numerous source files.

Each module will be formed by packages. Each package will have its own subpackages. For example editor will have kajax.editor package and in this package there will be subpackages like kajax.editor.utils, kajax.editor.test, kajax.editor.core.

Therefore, every kajax module name will start with "kajax." prefix.

3.1.2 Baselines

For our **kajax** IDE, we are working parallel and for the period of development we need some self-consistent baselines of the whole program. These baselines will be used as checkpoints of the development process. CVS server will be used to physically define these baselines. To define a snapshot of the development product as a baseline, a group consensus is needed. Testing will also be organized on these baselines. That is to apply testing the baseline identification will be processed.

3.1.3 Documents

Throughout the implementation process our group members will be preparing several documents such as CMP, Testing Specification Sheet, Installation Manual and User's Manual. The naming convention for documents is "krnc_" followed by document name. all of our documents name is in lower case and all words in a document name are separated by underscore ("_"). An example is: krnc_installation_manual.pdf.

3.2 MANAGEMENT AND CONTROL

Although we have done the task assignment of modules for the implementation considering the similarity of modules between each other, assignment of one team member can do an effect on other team member assignment when we look at the structure of our project. For example, most of the operations done on the modules such as database, solution explorer, toolbox have a result on our editor module so the probability of causing a change between these modules is very high during our implementation process. So below we will explain the procedure of how the changes will be handled among group members.

3.2.1 Change Request

- If the change request is coming from the assistant, Caner will deal with this request; talk the details of the change with the assistant. If the change request

is coming from one of the members of the team, this team member is responsible for discussing the details of the change with Caner and informing Caner about the nature of the change.

- With the information given by the change requester, and if the change affects more than modules Caner arranges a meeting for discussing the approval of this change. If the change is local it's enough to have approval of each member via e-mail.

3.2.2 Meeting

- Before the meeting Caner will ask other members for their thoughts about the sensibility of the change. If it is found sensible then the meeting time will be arranged. Before the arranged meeting the effects of the change must be determined by the change requester.
- During the meeting a list of tasks will be constructed for handling the change. This list will also be a validation check list for monitoring the change.

3.2.3 Change Request Evaluation

- An estimation of implementation time deviation from the previously constructed schedule will be prepared.
- Technical availability of the change will be discussed.
- Project manager will evaluate the change in the overall project progress aspect.

	CONFIGURATION MANAGEMENT PLAN	Date: 11.03.2007
---	--	------------------

- Decision for the approval of the change will be taken.
- If the change is approved, the tasks are assigned to members.
- Every member reports the progress of his task to Caner.
- Caner will report any problem about the progress of change implementation to project manager.
- Change will be replaced to CVS server.

3.2.4 Defect Tracking

Defects are observed by Caner during testing and by the other members during implementation.

- If a problem is found it will be reported to Caner.
- If the number of problem reaches to previously determined limit Caner will arrange a meeting for discussing these problem.
- During the meeting each problems will be discussed and a set of problems will be set as defects.
- For correcting the defects, the tasks are prepared and assigned to members.

3.2.5 Version Control

	CONFIGURATION MANAGEMENT PLAN	Date: 11.03.2007
---	--	------------------

Version numbers will be taken from the CVS server, each version will also have a text name.

3.3 CONFIGURATION STATUS ACCOUNTING

As the code implementation process proceeds, there will be several commits to the CVS repository. To prevent problems, each commit will be commented. Moreover, each group member will hold a list of individual commits with file name committed, date and comment. These lists will be merged to trace the process and backtrack in case of a bug.

3.4 CONFIGURATION AUDITING & REVIEWS

Configuration auditing will be used to achieve program quality and integrity. Once in two weeks, a meeting for auditing will be organized. In this meeting, each member will tell the others what he has added to the baseline configuration. The members will discuss whether the implementation is consistent with the design and will decide to request a change if needed. As the modules are completed, a final verification for this module will be made and that module's task will finish.

Moreover, each member will tell about the problems he has faced during the implementation and will ask for advice. A team member who has an idea to go through that bottleneck will provide documentation and personal help as soon as possible.

4. PROJECT SCHEDULES

We have prepared a living-schedule and put it on our web site. It will be updated regularly considering the time constraints and the remaining work. We are conducting regular meetings every week. One of the meetings will be among the team members and another will be between assistant of the project and our team members. In the first meeting which is held among members, we will analyze and

discuss the change requests related to the whole project and we will try to find solutions to the problems during the implementation. And also we will rearrange the project schedule and progress. In the second meeting which will be held between the assistant and team members, we will review the finished modules of the project and will take advises of the assistant about the finished and ongoing parts of the project.

We have CM milestones which are the deadlines of audits of the project. According to these milestone dates which are announced in the living schedule, we will arrange our progress and implementation of the program.

5. PROJECT RESOURCES

It is necessary to have some tools, techniques, equipments, and training to make CM activities. We will use the following tools in our project progress:

- CVS(used for holding and updating our products)
- FTP(used for updating our project's website and inform the progress)
- Web Server(used for the location of the website)

6. PLAN OPTIMIZATION

For the development progress we have the flexibility for changing the CM plan. Especially for risk management the CM activities and procedures required to process them could be simplified. The overall development cycle and progress will be monitored by the project manager. That is a plan optimization request can only be made by Okan (Project Manager). But this request can be granted by group consensus. The two foreseen sources for plan optimization need are:

- The development is behind the planned schedule.
- Different procedures are necessary for the safety and the usability of the development products.

