

W-eXpert

FREELANCERS

CONFIGURATION MANAGEMENT PLAN

1394659 Serhat ALYURT

1395508 Ahmet Kutlu ŞAHİN

1448869 Caner KAVAKOĞLU

1448851 Yağız KARGIN

TABLE OF CONTENTS

1. INTRODUCTION.....	3
1.1 PURPOSE OF THE CONFIGURATION MANAGEMENT PLAN.....	3
1.2 THE SCOPE OF THE CONFIGURATION MANAGEMENT PLAN.....	3
1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	3
1.4 DOCUMENT REFERENCES.....	4
2. CM FRAMEWORK ORGANIZATION.....	5
2.1 THE ORGANIZATION OF SCM TEAMS	5
2.2 THE RESPONSIBILITIES	6
2.3 TOOLS AND INFRASTRUCTURE.....	6
3. CONFIGURATION MANAGEMENT PROCESS.....	8
3.1 IDENTIFICATION.....	8
3.1.1 Source Code.....	8
3.1.2 Data.....	8
3.1.3 Baseline.....	9
3.1.4 Documents.	9
3.2 CONFIGURATION MANAGEMENT AND CONTROL.....	10
3.2.1 Requesting Changes	10
3.2.2 Evaluating Changes.....	11
3.2.3 Implementing Changes	12
3.3 CONFIGURATION STATUS ACCOUNTING.....	12
3.4 AUDITING	13
4. PROJECT SCHEDULES AND CM MILESTONES	13
5. PROJECT RESOURCES.....	15
6. PLAN OPTIMIZATION.....	15

<i>FREELANCERS</i>	CONFIGURATION MANAGEMENT PLAN	<i>W-eXpert</i>
---------------------------	--	-----------------

1. INTRODUCTION

With this document we, Freelancers, want to describe our Configuration Management Plan for our project, [W-eXpert](#).

1.1 Purpose of the Configuration Management Plan

“Software is not maintained, it is developed!” (CENG350 Software Engineering course 2007-spring Assoc. Prof. Dr. Ali H. Doğru). Within this scope, like every developing technology, our project also needs lots of changes, upgrades, debugging etc. However these changes must be informed to all members of the team and must be done under some strict rules. Because without rules there will be chaos, especially with long lines of codes. A little change even on a parameter of a function will absolutely bring problem to the other members of the team. To prevent these problems, we will need a repository and a tool (CVS, SVN etc.) for comparing between the code in repository committed by other team members and our code in our PC.

Other than these, we have divided our team into roles for the configuration management, which will be described in this document later.

1.2 Scope of The Configuration Management Plan

As we have mentioned above our project needs a configuration management plan which will maintain the required changes without any problems. We need this plan because our project has many modules interacting with each other and any changes in one will need to be done according to this plan. Also with this plan, making configurations on the project and watching over the ongoing configurations by other team members will become easier. This Configuration Management Plan will help our team to develop our project in a more organized manner.

1.3 Definitions, Acronyms and Abbreviations

This part of the Configuration Management Plan gives some definitions, acronyms and abbreviations:

ACRONYMS	DEFINITIONS
CCB	Configuration Control Board
CM	Configuration Management
CI	Configuration Item
SCM	Software Configuration Management
CVS	Concurrent Versioning System
CMP	Configuration Management Plan
SDT	Software Development Team
SCR	System Change Request

1.4 Document References

In this part we will give references that we get help preparing this Configuration Management Plan:

- IEEE Standard for Software Configuration Management Plans, IEEE Std 828 – 1998
- IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
- IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.
- IEEE Std 1042-1987 (Reaff 1993), IEEE Guide to Software Configuration Management.

2. CM FRAMEWORK ORGANIZATION

2.1 The Organization of SCM Teams

As Freelancers team members, we have formed four organizational units related to the SCM activities in the w-eXpert using old Ceng490 project group's experiences.

- Change Evaluation Team (CET)
 - Caner KAVAKOĞLU
 - Serhat ALYURT
- Configuration Management Team (CMT)
 - Caner KAVAKOĞLU
 - Yağız KARGIN
 - A. Kutlu ŞAHİN
- Testing Team (TT)
 - A. Kutlu ŞAHİN
 - Serhat ALYURT
 - Yağız KARGIN
- Software Development Team (SDT)
 - Serhat ALYURT
 - Yağız KARGIN

<i>FREELANCERS</i>	CONFIGURATION MANAGEMENT PLAN	<i>W-eXpert</i>
---------------------------	--	-----------------

- Caner KAVAKOĞLU

- A. Kutlu ŞAHİN

We interiorized the decentralized democratic team model.

2.2 The Responsibilities

- Change Evaluation Team (CET): This team is responsible for the evaluation 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 will be responsible for assuring the quality of the end product through avoiding the propagation of the errors between implementation stages. Preparing the test cases and use cases, extensive inspection of the individual modules through unit tests and publishing the results will be the main duties of TT.
- Software Development Team (SDT): This team will work for implementation of w-eXpert. Team members will integrate the accepted change request into the required module specified by CCT.

Because of the fact that we interiorized the decentralized democratic team model, we arrange a meeting about any issues except missions.

2.3 Tools and Infrastructure

- We use CVS and Eclipse.

CVS: We determined the best solution of directory problem that is about which document will be stay under which directory. Below we show the directory tree and their contents;

w-expert : Project node

images : GUI related images

src: all w-expert related source codes

classes : w-expert modules and their implementations

api_module

general_module

im_module

profile_module

search_module

db : sql files

lib : java (i.e. smack library), javascript libraries

tmp : test aimed codes

web : client side codes

css

jsp

js

3. CONFIGURATION MANAGEMENT PROCESS

3.1 Identification

In order to identify the project we have the following parts which are; source code, data, baseline and documents.

3.1.1 Source Code

It is obvious that the part which will be subjected to change is the source code of the project. So, to maintain the integrity and modularity of the project's source we have arranged the project structure so that it has five modules namely:

- General Module
- Profile module
- Search module
- Instant messaging module
- API module

We will be using jsp, Java, and CVS for a coordinated work. We will commit the source codes that we write every week under the src file of the CVS repository. The total hierarchy of the files is mentioned in the previous sections. The file named 'classes' which is under the src file, will have the classes that we write.

3.1.2 Data

This data part of the project will consists of the GUI related sources. They will be some kind of images, icons etc. so the visualization will be easy to be replaced, with the help of a style sheet.

3.1.3 Baselines

The baselines consist of the milestones of the project. It is an important criterion because these milestones or checkpoints will show the current progress of the project. To achieve this integrity we will use CVS and commit the sources every week and check the system as a whole. Finally, baselines will be tracked and version controlled in CVS by respective CM teams. The main baselines for the team will be

- Requirements analysis
- Initial design
- Final design
- Configuration management plan
- Implementation and integration
- Version 1.0
- Version 1.1
- Testing
- Documentation
- Final presentation and demonstration

3.1.4 Documents

For a good team work our company, Freelancers, have published several documents. These documents can be used by a person outside the team to track our project from proposal to the design. And these documents are also essential for the team members to maintain the coordination and to handle such a project.

The documents that we have published are:

- Project proposal
- Requirements analysis report
- Initial design report
- Final design report
- Living schedule
- Configuration management plan

3.2 CONFIGURATION MANAGEMENT AND CONTROL

Although we divided our project into modules which has minimal effects and overlapping parts between each other, sometimes a change in one of the modules can negatively affect other modules of W-Expert. To make that negative effect minimal, we have formed a changing procedure. In this part of the CMP we have explained that procedure, namely how these changes will be handled by Freelancers.

3.2.1 Requesting Changes

Some of the team members may come up with CRs due to some reasons, while the development process of W-Expert is going on. These reasons may be a small mistake in design or implementation that has to be corrected or may be a new design decision which can take us to an easier solution of the existing problems of W-Expert. Two ways can be used while making a CR. The first way is e-mail. The owner of the CR writes an e-mail and sends it to all members of or team. We has a template for mailing for this issue. This mail consists of

- date and time of CR,
- deadline of CR,

- change requested module name,
- current version ids and names of source files
- priority of CR
- brief description of CR.

Some of these items must exist in CR mail. These are date and time of CR, because this will be used as the id of the CR, module name, version id and name of the source files, because one should know where the change is required before discussing it. The second way is telling the CR in our weekly meetings. It will be taken as a note. Also this note must include what a CR e-mail contains.

In addition our instructor or assistant can make CRs. These CRs can be in any form and they will be taken as notes like the CRs from the team members.

3.2.2 Evaluating Changes

Although changing and maintaining are usually more difficult than restarting, sometimes small changes which correct the mistakes or make our work easy are acceptable and useful. However, most of the time they are difficult to implement, so difficult to evaluate. Because of these reasons a CR should be considered and evaluated realistically, but not superficially. After thinking about these issues, we have decided to judge a CR extensively. When a CR is occurred, it does not matter which way it is submitted, we will discuss it in the next weekly meeting. If the priority of the CR is high, i.e. change should be made urgently, we will not wait for the next meeting. We will arrange a new meeting if we are suitable for that or arrange a conversation via Internet services (e.g. Skype). Before the arranged meeting the effects of the change must be determined by the change requester. During the meeting or conversation we discuss the terms of implementing that requested change. We generally agree with the change when it is local and not have side effects. However we know usually this is not the case that we will face. Therefore at the meeting, effects on overall project progress and technical availability will be discussed. Implementation time will be estimated and plans for the coming weeks will be changed accordingly.

3.2.3 Implementing Changes

If the CR is approved in the meeting or conversation, one or more members - according to the effect size - will be assigned to the implementation of changes. This person will usually be the person who has made the CR. Then, the responsible person fixes the code according to the way specified in the meeting and uploads it to the CVS. Thus, others member can reach the latest updated version of the module.

Defects can be observed by Yağız while auditing or by other members while testing or by the member who implements the change. If the defects size is large, this will cause a new CR to be made and a meeting to be arranged. For correcting the defects, the tasks will be prepared and assigned to members. They are especially assigned to the requester of the change which cause the defect, because he will be the most familiar person to changed part of the project.

3.3 CONFIGURATION STATUS ACCOUNTING

This part is mainly about recording, knowing and reporting the status of W-Expert. By CSA, we will report to all the group members about the status of the project. Since we discuss the terms of the changes all together, every member will have the required information about the status. At the first semester, members were informed about the status at the weekly meetings. The second semester this will continue like that. In addition to weekly meetings, we are using CVS this semester. CVS will show a member where we are in the completion of the project. Also assistant and instructor can learn the progress of the project by examining the CVS. To prevent problems, each commit will be commented and codes should also be commented. So, when a member puts his source code to CVS, the related document or comments should be provided by him/her. Furthermore, our assistant can learn the status of W-Expert from our weekly reports as he can in the first semester. Besides all those explained above, the status of the project will be seen in the living schedule updated weekly in the web site.

3.4 AUDITING

Auditing is a very vital issue of software configuration management. It makes our project appropriate and favorable. Also its main purpose is to make the project meet the requirements specified in the analysis phase. Auditing will be done mainly by Yağız, however every member should have audit the progress of the project. If a change is required, it can be done only with a CR, following the procedure explained above. We believe that auditing is very important and Yağız can handle it with the help of the others.

We will do three types of auditing. While doing functional audits, our main purpose is to verify the functionality of the configuration items. Some functional requirements that have to be verified and which member they are assigned are:

- the proper functioning of database and CVS (Caner)
- the proper functioning of menu bars, links, styles etc. in the web pages (Kutlu)
- the proper functioning of instant messaging and expertise matching (Yağız)
- the proper functioning of debugging and testing process (Serhat).

Although these functional audits are assigned to specific members, they are fully separated from each other. This means a member can audit parts that are not assigned to him. Yağız will be auditing the development of the project generally.

With physical audits we will examine if the specifications and standards applied are correctly addressed and can be observed through technical requirements.

We will do process audits to observe and verify the reliability, integrity and consistency of the development process. We will examine the progress of the development stage and we can make some decision about the planning according to this examination.

4. PROJECT SCHEDULES AND CM MILESTONES

In this part of the CMP we will try to give our schedule for our project, information about the sequence of tasks and coordination of the CM activities. Also our living schedule can be seen on our website which is updated every week (till Saturday morning).

On every Wednesday morning we have a meeting with our assistant Çağatay Çallı and we discuss our project's process. With these discussions and the schedule in our hand we will proceed in our progress more planned.

Other than the meeting with Mr. Çallı our team will come together every weekend to discuss the jobs that have been done this week and what needs to be done next week. With these meetings we aim to increase our teams motivation and determination. Also found bugs and problems can be discussed and solved within these meetings.

To give a more detailed schedule we defined some milestones for our Configuration Management. A little differences in the given dates can be viewed from our Living Schedule in our website (<http://senior.ceng.metu.edu.tr/2008/freelancers/images/gantt2.JPG>):

- Chat Module Implementation (15.03.2008)
- Profile Module Implementation (20.03.2008)
- General Module Implementation (30.03.2008)
- Search Module Implementation (05.04.2008)
- Integration of Modules (7.04.2008)
- First Testing and Release Version 1.0 (08.04.2008)
- API Module Implementation (20.04.2008)
- Finalizing Integration of Modules (25.04.2008)
- Second Testing and Release Version 2.0 (08.05.2008)
- Debugging and Improvements on GUI and Modules (30.05.2008)
- Documentation and the Final Release (12.06.2008)

5. Project Resources

To maintain a solid team work we have to use some tools. In this case it is CVS. Without using the CVS it would be hard to reflect the changes to the project that are done by 4 team members. So there will be no conflict after an update with the help of this versioning tool. It will be easy to access the latest version and make some updates concurrently.

Another resource is the web site of the w-expert, which includes the available latest versions of the w-expert, documents of the project and living schedule.

6. Plan Optimization

Our team members will strictly obey the deadlines and follow the milestones of the project. We will share the work in every implementation step, keep following the structures and classes described in the final design report. For every week we will commit the new source codes to the cvs repository and after we agree on that version, we will start with the next step.

Meanwhile we will be discussing the process in our weekly group meetings.