



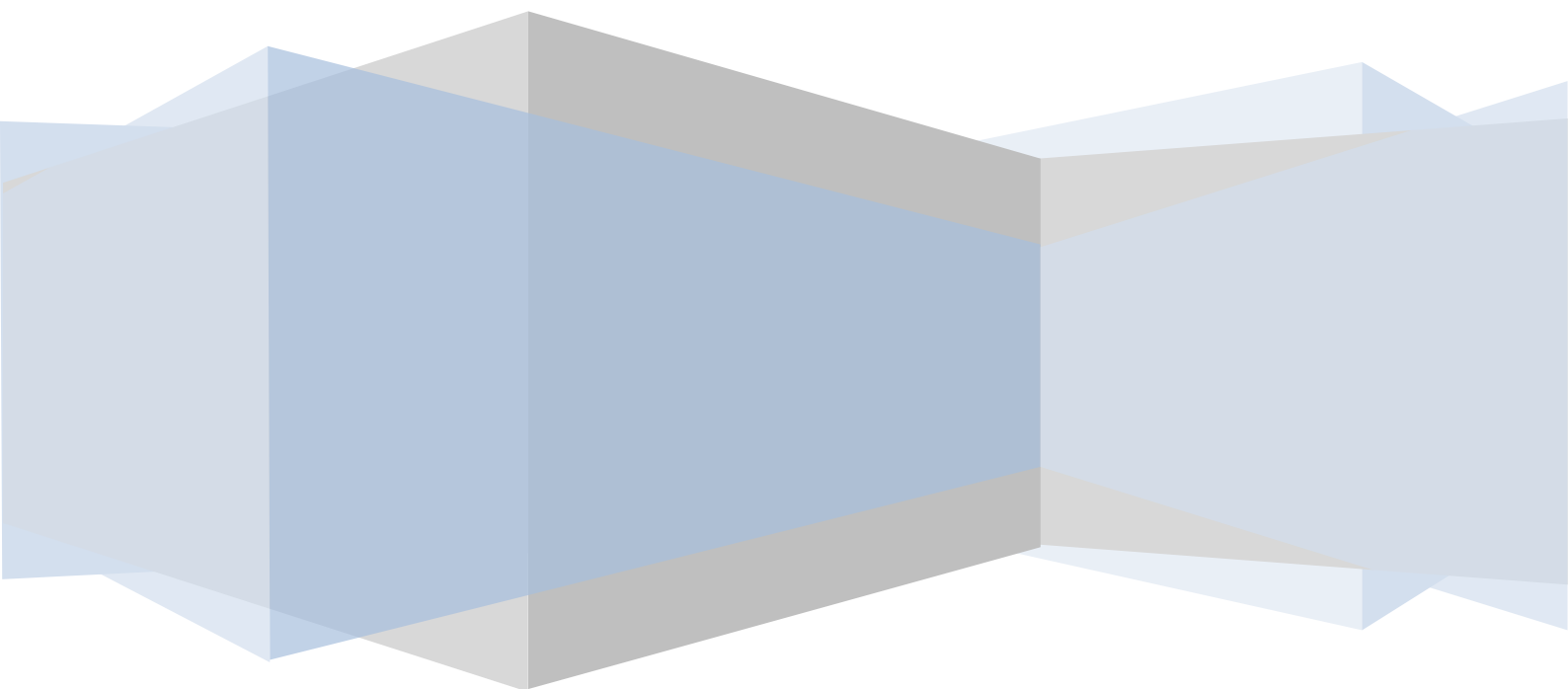
PseudoSoft

CENG-492

CONFIGURATION MANAGEMENT REPORT

YENILINK PROJECT

Assistant: Ali Orkan Bayer



GROUP MEMBERS

Name	Number	E-mail
Furkan Kürşat Danışmaz	1394881	k.furkan@gmail.com
Ömer Nebil Yaveroğlu	1449248	omernebil@hotmail.com
Mehmet Bahattin Yaşar	1395664	e1395664@ceng.metu.edu.tr
Gülsüm Selcen Mülazımoğlu	1395276	selcen.mulazimoglu@gmail.com

Table of Content

1. INTRODUCTION	2
1.1. Purpose of CMP	2
1.2. Scope of Document	2
1.3. Definitions, Acronyms, and Abbreviations	3
1.4. Document References	3
1.5. Document Overview	3
2. THE ORGANIZATION & CM FRAMEWORK	4
2.1. Organization	4
2.2. Responsibilities	5
2.3. Tools & Infrastructure	5
3. THE PROCESS OF CONFIGURATION MANAGEMENT	5
3.1. Identification	5
3.2. Management & Control	7
3.3. Status Accounting	7
3.4. Auditing	7
4. PROJECT SCHEDULE AND CM MILESTONES	8
5. PROJECT RESOURCES	9
6. PLAN OPTIMIZATION	9
APPENDIX	10

1. INTRODUCTION

1.1. Purpose of CMP

The purpose of this software configuration management (SCM) plan is to maintain the continuity of our project "Yenilink". Since the change is inevitable during development of all software projects, and our project is not an exception, making modifications should be easy enough in our project. In addition, the possible modifications should not confuse any one of our project group members. Thus, this configuration management plan is prepared to define the process of identifying, managing, and auditing the changes as they occur throughout the lifecycle of "Yenilink" Project.

1.2. Scope of Document

The scope of this document is to define and explain all the configuration management properties of Pseudosoftware's Yenilink project. The activities discussed here are applicable to all documentation, source code development, software and hardware tools used, and any other process involved. Thus, this document determines the responsibilities and authorities for accomplishing the planned activities, details of the items under configuration management process and necessary coordination of SCM activities with the other activities in the project. Using this Configuration Management Plan makes recognition, reexamination, and identification of items of software configuration of the project unambiguous to all of project team members. We can clarify the control of changes of codes and we can describe more realistic software configuration in terms of time. In other words, this report is intended for the group members in first place. Our supervisor and the instructor are among the other audience of the document.

1.3. Definitions, Acronyms and Abbreviations

JSWS: Job Seeking Web Site

TMP: Temporary

GUI Package: Graphical User Interface Package

WS Package: Web Services Package

Docs: Documentations

CM: Configuration Management

CMP: Configuration Management Plan

SCM: Software Configuration Management

1.4. Document References

While preparing this plan, we used the following documents as a reference:

- Software Engineering: A Practitioner's Approach – Roger S. Pressman, 4th Edition
- Software Configuration Management Plan, Presentation prepared by METU Computer Engineering Department for the course Ceng 492
- Our Requirement Analysis Report
- Our Final Design Report

1.5. Document Overview

This document consists of six main parts, each of which is described below:

- **Introduction:** In the introduction part, we explained the purpose of CMP. On the other hand we explained the scope of the document, definitions and abbreviations, references.

- **The Organizations CM Framework:** In this part, we determined the organization and responsibilities of all team members for CM. Moreover, we explained the infrastructures and the tools we will use in configuration management.
- **The CM Process:** The identification, management, and the auditing of the (CIs) are discussed here.
- **Project Schedules and CM Milestones:** The deadlines for the CM activities are given in this part.
- **Project Resources:** In this part, we explained the project resources which will be necessary for CM.
- **Plan Optimization:** This section explains the methods that can be used to optimize the CMP.

2. THE ORGANIZATION & CM FRAMEWORK

2.1. Organization

Our group consists of four people. There is a good communication between our group members, since each of us has a complete understanding of how things are to be handled. On the other hand, to carry on the Yenilink project successfully, we have grouped the tasks to be fulfilled into categories. These categories can also be seen as our modules. These are mainly “Databases”, “User Interfaces” and “Web Services”.

Depending on these categories, we have planned our CVS structure. You can see a graphical representation of our CVS directory structure in the Appendix part. We have designed this in a way that goes parallel to our development process. We more or less complete our tasks module by module so keeping a directory structure based on modules seemed more suitable for us. The updates and changes will only affect one of the modules which are being worked on and others will not be affected. We have also divided these modules into the different parts of the project like portal, JSWS and by this way a change made will not affect another part of the project. If we have made the development based on these parts of the project, we may think about another directory structure in which the first

division was based on modules and the second division was based on modules. But this directory structure didn't seem suitable for our aim.

2.2. Responsibilities

We have mentioned our CVS folder structure in the "Organization" part of this report. Now in this part of the report, we will mention who is responsible for which part of this tree. By responsibility, we mean both development related responsibility and transfer of the developed code to CVS responsibility. Also keeping the directory tidy and avoiding any mess is also important.

- SQL, JSWS1 and JSWS2 directories under DBPackage directory → Bahattin
- Portal directory under DBPackage directory → Furkan
- Portal directory under GUIPackage directory → Ömer + Selcen + Furkan
- Bank, JSWS1 and JSWS2 directory under GUIPackage directory → Ömer + Selcen
- All the unmentioned left parts → All group members

Apart from these responsibilities, Ömer is responsible for informing an important or big change to group members.

2.3. Tools & Infrastructure

- Eclipse CVS in the design of Web Services and GUI Modules
- NetBeans CVS in the design of Database Modules

3. THE PROCESS OF CONFIGURATION MANAGEMENT

3.1. Identification

There are three main parts of our project which are "the Database", "Graphical User Interface", and "the Web services" as we use the "Three Tier Architecture". Each of these three is being developed as independent projects for now.

The Database: Apart from the database structure and sql scripts (the create statements), we have our database utility modules containing insert, delete, update, and select methods implemented with Java using Hibernate technique. The database directory will contain three subdirectories as we have three different databases (one for portal, and two for two different Job seeking Website we are simulating).

And each subdirectory will contain four subdirectories under “src”:

- **globalPack**
- **independentTablesPack**
- **utilityPack**
- **HibernateTesing**
- **SQL**

“globalPack” contains “SessionUtility.java” which is used by every hibernate utility class. “independentTablesPack” contains Java class and hibernate mapping file again for every table for every table in the corresponding database. “utilityPack” contains different Java classes for each table containing insert, delete, update, and select methods. Finally, “HibernateTesting” package contains different Java classes for each utility class. This testing package will be taken out in the release versions of the Project since it contains nothing usefull but codes for testing.

The Graphical User Interface: This directory contains four different subdirectories (apart from tmp) as we have three different Websites to be implemented (one for our portal, and two career Websites and one for bank).

The Web services: This directory contains all the Web services. Again we have three different subdirectories (one for portal, and two for each career Website) apart from “Tmp”. Each Web service is independent of the other to be deployed in the server. Therefore we will have different subdirectories for each service under corresponding directory.

SQL: Finally the SQL directory as a final subdirectory under “Database” includes the sql scripts for creating the databases and tables for each database.

3.2. Management & Control

We have divided our work into fragments so that the dependence of one developer to another is minimized. We take all the decisions together, but the development of each module is under different person's responsibility. Therefore, there will be no difficulty on versioning the modules. Anyone can upload the new version of a module without bothering the others. The only thing to be done while versioning is informing the other member's about the new version of the module so that the integration of the module takes place immediately. As new modules are added to the system, their integration and testing will be done and if the changes cannot be applied successfully, the new version of the module will be rejected.

3.3. Status Accounting

Every member is responsible for preparing "readme" file for each module, infact for anything that is under his/her responsibility. For every change, there will be explanation about what change is made and why. For such a big Project, the reasons for the decisions could easily be forgotten. Therefore, every change and every decision should be documented with all the reasons why they are applied. Apart from the changes and the decisions, the bugs will also be reported.

3.4. Auditing

As any new module is developed or a new version of any module is uploaded, it will be tested and the decisions about it will be taken by all the members together. Although the dependency of development is minimized, the group members are still dependent to each other. As a one example, the select methods of the database utility classes should be implemented according to the user interface decisions. Therefore the auditing will be done together as new modules, or new versions of existing modules are uploaded.

4. PROJECT SCHEDULE AND CM MILESTONES

We have made some changes in our project schedule during the preparation of living schedule. These changes were not made because the schedule was unrealistic but it was made because we have prepared it in so much detail. These details made our living schedule difficult to follow on a day-by-day basis. Also there were some completed parts in the first term although they were in the scope of the second term. There were also some parts which were not done although it was planned to be made in the first semester. These differences have occurred since the flow of the development process lead us that way. So we updated and generalized our schedule. You can reach our schedule from “Appendix” part of this report.

Apart from our project schedule we have defined some milestone for our project development. There are demo days defined by our assistant (first one is on 1 April and second one is on 6 May). There is also one final demo on 13 June. These demo days rules us a lot. Taking these demonstration days into account we have defined our milestones as below:

- | | | |
|----------|---|--|
| 17 March | → | All database implementations will be completed. |
| 1 April | → | Some of the web services will be implemented. |
| | → | Most of the user interface requirements will be implemented. |
| 28 April | → | All web services will be implemented basically. |
| 6 May | → | All user interface parts will be implemented. |
| | → | Some optimizations will be made. |
| 1 June | → | All web services will be completed in detail. |
| 13 June | → | All parts will be integrated and tested to work together. |

5. PROJECT RESOURCES

Up to this day, we have managed our codes and documents using Groove and manual transfers. But the project is getting bigger and the importance of our project files increase. Although we take many backups on DVD's and hard disks, it doesn't seem reliable enough to go on this way. Since we are provided by CVS accounts, we will use CVS as our configuration management tool. We plan to use Netbeans or Eclipse IDE as our connection way to our CVS account. These two IDE are quite similar to each other by means of CVS usage. The decision will be made depending on the IDE we use at the time of submission. Although we use CVS by one of these ways, we will continue to backup our code using Groove and disks.

We have weekly meetings and share the new thing we have made or discovered. We also communicate with each other using Groove and MSN Messenger when we need immediate access. By one of these ways, we will inform about the changes on CVS.

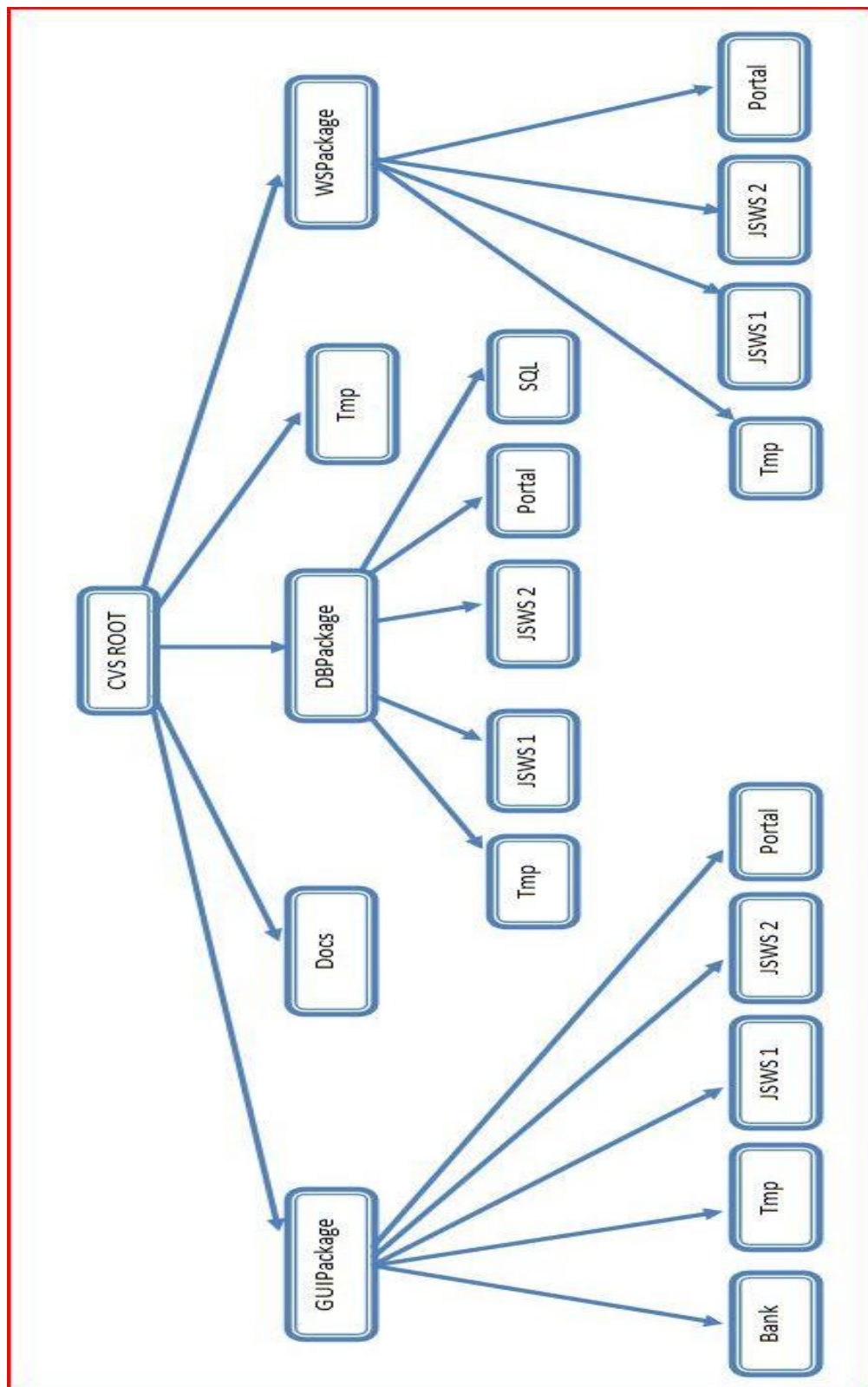
We will create our documentation about the project using "Wordpad" since we plan to include our documentation files on CVS. This will give us the flexibility of opening the documentation files with any of the word processors existing in any computer.

6. PLAN OPTIMIZATION

Although we have written our configuration management plan here, we are sure that there will be slight changes on the dates and orders of the process to be done since it is done on predictions. We will try to stick to the configuration management plan during our development process but we may need some small changes in this plan. These changes will be welcomed by our group if it doesn't cause a big problem in our demos. We have even changed our schedule since it became out of data after a while. So changes are vital and we will optimize the plan in a way that there will not be any problems in our demonstrations.

APPENDIX

1. OUR CVS DIRECTORY STRUCTURE



2. OUR SCHEDULE

	Task Name	Duration	Start	Finish	Resource Names
1	☒ Learning and Practicing Development Tools	100 days?	Mon 15.10.07	Thu 28.02.08	
2	Tomcat & Axis 2	24 days?	Mon 15.10.07	Thu 15.11.07	Everyone
3	Flash Development	14 days?	Mon 11.02.08	Thu 28.02.08	Everyone
4	J2EE Topics & Web Services	54 days?	Fri 02.11.07	Tue 15.01.08	Everyone
5	Hibernate	11 days?	Fri 01.02.08	Fri 15.02.08	Everyone
6	Taking Design Considerations	70 days?	Tue 16.10.07	Fri 18.01.08	Everyone
7	☒ Design Of Databases	51 days?	Mon 12.11.07	Fri 18.01.08	
8	Database Design Of JSWS1	51 days?	Mon 12.11.07	Fri 18.01.08	Everyone
9	Database Design Of JSWS2	51 days?	Mon 12.11.07	Fri 18.01.08	Everyone
10	Database Design Of Yenilink Portal	51 days?	Mon 12.11.07	Fri 18.01.08	Everyone
11	User Interface Design	26 days?	Sat 15.12.07	Fri 18.01.08	Everyone
12	☒ User Interface Implementation	71 days?	Fri 01.02.08	Fri 09.05.08	
13	User Interface Implementation Of JSWS 1	71 days?	Fri 01.02.08	Fri 09.05.08	Ömer Selcen
14	User Interface Implementation Of JSWS 2	71 days?	Fri 01.02.08	Fri 09.05.08	Ömer Selcen
15	User Interface Implementation Of Yenilink Portal	71 days?	Fri 01.02.08	Fri 09.05.08	Ömer Selcen
16	☒ Implementation Of Databases With Hibernate	20 days?	Mon 18.02.08	Sun 16.03.08	
17	Implementation Of JSWS1	20 days?	Mon 18.02.08	Fri 14.03.08	Bahattin
18	Implementation Of JSWS2	20 days?	Mon 18.02.08	Fri 14.03.08	Selcen
19	Implementation Of Yenilink Portal	20 days?	Mon 18.02.08	Fri 14.03.08	Furkan
20	Testing Databases	20 days?	Mon 18.02.08	Sun 16.03.08	Everyone
21	☒ Implementation Of Web Services	33 days?	Mon 17.03.08	Wed 30.04.08	
22	Implementation Of JSWS1	33 days?	Mon 17.03.08	Wed 30.04.08	Bahattin
23	Implementation Of JSWS2	33 days?	Mon 17.03.08	Wed 30.04.08	Furkan
24	Implementation Of Yenilink Portal	33 days?	Mon 17.03.08	Wed 30.04.08	Selcen
25	Testing Of Web Services	33 days?	Mon 17.03.08	Wed 30.04.08	Everyone
26	Integration Of All Components	11 days?	Thu 01.05.08	Thu 15.05.08	Everyone
27	General Testing and Debugging	21 days?	Thu 15.05.08	Thu 12.06.08	Everyone