

ASPIRIN SOFT

Online CV Portal
(OCP)

CONFIGURATION
MANAGEMENT PLAN



PROJECT ASSISTANT :

Ali Orkan BAYER

TEAM MEMBERS:

Bayram ERSOY

Emre ARSLAN

Gökhan ÇETİN

Berkay DEMİRER

Table of Contents

1.	Introduction.....	3
1.1	Purpose of CMP.....	3
1.2	Scope of the Document.....	3
1.3	Definitions, Acronyms and Abbreviations	3
1.4	Document References	4
2.	The Organizations CM Framework	4
2.1	Organization.....	4
2.2	Responsibilities	4
2.3	Tools & Infrastructure.....	5
3.	The CM Process.....	6
3.1	Identification	6
3.1.1	Code:.....	6
3.1.2	Data:.....	6
3.1.3	Documents:	6
3.2	Management and Control.....	7
3.2.1	Change Requests	7
3.2.2	Evaluation of change requests	7
3.2.3	Implementation of change requests	7
3.3	Configuration Status Accounting.....	7
3.4	Auditing	8
4.	Project Schedules	8
5.	Project Resources	9
6.	Plan Optimization	9

1. Introduction

1.1 Purpose of CMP

This management plan is written for the Online CV Portal project of the software development team Aspirin Soft. In software development process countless changes are made at every stage, since software developer's understanding of the problem gets better as she/he makes progress in the solution, his/her approach to the problem solution also changes. This brings the need to make changes to the software. Managing and handling these changes is another issue for the development team. In this management plan we will specify our strategy to make these changes. This strategy ensures that the changes made to the project and the versions do not affect Online CV Portal's completeness.

1.2 Scope of the Document

In this document the configuration management plan for AspirinSoft 's Online CV Portal project is presented. The organization of the teams, their responsibilities and the actions to be taken are specified. The way the identification of the system, accounting and the audits are going to be done is explained in detail. This management plan aims to manage have full control over the changes done to the system and prevent any unexpected behaviour.

1.3 Definitions, Acronyms and Abbreviations

CM: Configuration Management

CMP: Configuration Management Plan

SCR: System Change Request

CMT: Configuration Management Team

DT: Development Team

TT: Testing Team

SQAT: Software Quality Assurance Team

1.4 Document References

-Software Configuration Management

Ceng 492 Computer Engineering Design II Spring 2008

<http://www.ceng.metu.edu.tr/courses/ceng490/documents/>

-Software Engineering: A Practitioner's Approach 5th Edition

Roger S. Pressman

2. The Organizations CM Framework

2.1 Organization

Since there are limited number of people working on this project there will be limited number of sub-groups and the configuration management will be done by the team itself. The sub-groups are;

- 1- Configuration Management Team: This team includes all the team members, since this is a small project carried out by only 4 people.
- 2- Testing Team: Testing team is Bayram Ersoy and Berkay Demirer
- 3- Development Team: Development team is Gökhan Çetin and Emre Arslan:
- 4- Software Quality Assurance Team: 1 member from the DT and 1 member from the TT is in this subgroup. Gökhan Çetin and Berkay Demirer

2.2 Responsibilities

- a. CMT: CMT is responsible for creating and maintaining the CMP. The coordination of CM activities and the reporting is also done by the CMT. This includes coordination of the subgroups also.
- b. DT: DT is the subgroup that implements and integrates the SCRs and the project itself.
- c. TT: TT's main responsibility is testing and reporting any bugs arising after the implementation of an SCR.
- d. SQAT: This team evaluates the SCRs. Maintains the system's completeness, guarantees that any change will not affect the project progress in a bad way. SQAT also checks if a change is really necessary and if all the possible options are taken into account while making the change. If the SQAT decides that a change must be done and the change request is the best option available it reports the situation to the DT for the change to be implemented.

2.3 Tools & Infrastructure

Eclipse & Application Designer as development environment.

CVS as version control system.

Oracle as our main DBMS.

MSSQL or MYSQL for external sites' DMBS.

Java as programming language.

Ms Windows Server 2003 as operating system on which the application runs.

Microsoft Internet Explorer and Firefox as client web browser.

3. The CM Process

3.1 Identification

Our configuration items are code, data and documents.

3.1.1 Code:

This project is a long term project so there will be many changes during implementation because we will make tests and change some actions or errors or add new functionalities. In order to cope with these changes we will implement many pages and actions, beans, page classes, etc. Each module will have its own directory and we will use CVS to update and commit changes. We will take weekly backups of our source code in order to get back to a working version if there is a serious problem.

3.1.2 Data:

We will deal with many data and we will have a huge database. In order to keep this data safe we will have weekly backups of our database .We will have database tables for keeping logs. Logs will be about actions of our clients and external sites.

3.1.3 Documents:

We prepared following documents for our project.

- Project Proposal
- Requirement Analysis report
- Initial design report
- Final Design Report
- Configuration Management Plan

We give importance to documentation and we will have new documents like “Test Plan” and “User Manual”. These documents will be accessible from our web site.

3.2 Management and Control

Since this project team has only 4 members, everything related to organization and reporting will be in an informal way. The back-ups, versioning and the storage of files and the naming of these will be done in a formal way for the sake of the project.

3.2.1 Change Requests

Change requests will be discussed in daily meetings or on the phone. Each member will have a working version of the application and he/she can make a change request informally to the group leader.

3.2.2 Evaluation of change requests

Whenever there is a change request about a module the reason for the change will be examined carefully. The owner of the module will think about the change and decide whether the change is appropriate or not. Finally group member will decide about the change after getting feedback from group members.

3.2.3 Implementation of change requests

If a change request is evaluated and seems ok a backup of the module will be taken and change will be implemented by the owner of the module. Comments about the change will be mandatory.

Finally all changes will be committed and group members will update the change.

3.3 Configuration Status Accounting

Each system change and each commit to the CVS repository will be logged by the team. Each system change will be listed with a name and a date. The date of a system change is going to be added to the end of the version number to be able to

do backtracking. For example if a system change is done at 20th of March to the 1st version of the Online CV Portal, the version name is going to be “V1.03.20”
Before any system change is done we will save back-ups of the system and when we need to backtrack we will look up the system change from the list and do the backtracking accordingly.

3.4 Auditing

Before a system change is done the SQAT team evaluates its necessity and if it will not spoil the system's completeness. The TT tests the system changes after the implementation but software audits are still essential. All team members take place in the audits and express his ideas about the change.

CMT is going to organize the audits. There will be once-in-2-weeks regular audits and there will be extra ones when CMT decides it's necessary.

For the Online CV Portal there are some measures for the usefulness of a change other than the completeness of the system. These are the system & data security, performance of the system, the change's adaptability to huge number of operations. If a change meets all these expectations it is confirmed if not the backtracking will be done after audits.

4. Project Schedules

Project Live schedule will be on our website soon.

Milestones of the project are as below:

Implementation of design of modules.	13.03.2008
Implementation of registration module.	15.03.2008
Testing and debugging of registration module	17.03.2008
Implementation of jobseeker module.	17.03.2008
Testing and debugging of jobseeker module	19.03.2008
Implementation of employee module.	19.03.2008
Testing and debugging of employee module	21.03.2008
Implementation of search module.	21.03.2008
Testing and debugging of search module	23.03.2008
Implementation of an external web site.	23.03.2008

Implementation of CV web services.	25.03.2008
Testing and debugging of CV web services	27.03.2008
Implementation of position available web services.	27.03.2008
Testing of position available web services	30.03.2008
Deployment and First Demo.	01.04.2008
Implementation of external sites.	10.04.2008
Implementation of notification module.	15.04.2008
Implementation of admin module.	20.04.2008
Implementation of statistics module.	30.04.2008
Deployment and Testing.	05.05.2008
Final Demo.	15.06.2008

5. Project Resources

We have to apply CMP and we have to follow CM activities so that our project has good value. Since our team consists of 4 people, it is important to work on the same source without having any problems. Therefore we are going to use CVS and have repository on the department machine. We will use Eclipse as CVS client.

6. Plan Optimization

The CMP is our guide in the development process. If we have some serious changes in the development process we are going to optimize the CMP and so our progress will not be broken. We have weekly meetings and also we have informal meetings during the week. As a result, it will be easier to communicate, decide and make necessary optimizations. We will have individual “metrik kayıt” files on the repository. All team members are responsible to report their updates, changes, workings on that file. By this way we will track our progress individually.