

# CONFIGURATION MANAGEMENT PLAN

***CoreSoft***

İREN BERK ÖZALP	1449040
ERDEM ALPAY	1448349
NURULLAH GÜRCAN	1448745
BELKAY SARATAŞ	1449081

# Table of Contents

1. Introduction .....	3
1.1 Purpose of CMP .....	3
1.2 Scope of Document.....	3
1.3 Definitions, Acronyms and Abbreviations .....	4
1.4 Document References.....	4
2. Organization of the CM Framework .....	5
2.1 Organization .....	5
2.1.1 Software Development Team .....	5
2.1.2 Configuration Management Team .....	5
2.1.3 Testing Team .....	6
2.1.4 Release Controlling Team .....	6
2.2 Responsibilities.....	6
2.3 Tools and Infrastructure .....	7
3. The CM Process .....	8
3.1 Identification .....	8
3.1.1 Code Modules .....	8
3.1.2 Documentation .....	8
3.2 Management and Control .....	9
3.2.1 Change Requests .....	9
3.2 Configuration Status Accounting .....	10
3.3 Auditing.....	11
3.4 Time Plan.....	11

## **1. Introduction**

With this documentation we try to give details of our configuration management plan.

### **1.1 Purpose of CMP**

In software project, development must be controlled with some predefined rules and conventions. Software projects can be finalized with step by step increments. In each step something will change and probably whole team, who is responsible for this project, shall not attend these changes, but all of the participants must be informed about changes. Also implementation is not the only important thing in a software project. Versioning, testing and supervising releases also must handled by software developing teams.

So, in order to avoid messes caused by the reasons explained before, we need to put a configuration management plan in our software development process.

### **1.2 Scope of Document**

In this documentation, we give information about our organizations to accomplish configuration management, separated roles for this task. The tools we plan to use are going to be described in this report with status accounting and auditing of the project. At last we will give a schedule of our project with milestones.

### 1.3 Definitions, Acronyms and Abbreviations

CM Configuration Management  
CMP Configuration Management Plan  
TT Testing Team  
DT Development Team  
PD Program Director  
RCT Release Controlling Team  
SVN Subversion  
SDT Software Development Team

### 1.4 Document References

IEEE Standard for Software Configuration Management Plans, IEEE Std 828 – 1998

“Detailed Design Report”, Coresoft

Pressman, Roger S. (2001). Software Engineering: A Practitioner's Approach, Fifth Edition. New York, NY: McGraw-Hill

“Configuration.Management.ppt”, presentation provided by METU CEng Department

[http://en.wikipedia.org/wiki/Subversion\\_\(software\)](http://en.wikipedia.org/wiki/Subversion_(software))

## **2. Organization of the CM Framework**

### **2.1 Organization**

We divide our organization into 4 parts. For every part a team is responsible for that part's tasks. We will use decentralized democratic team model. Because we have a little number of developers with almost same knowledge and experience, also our project is suitable for this kind of organization.

#### **2.1.1 Software Development Team**

Software Development Team consists of 4 members;

İren Berk Özalp

Nurullah Gürcan

Erdem Alpay

Belkay Sarataş

#### **2.1.2 Configuration Management Team**

Configuration Management Team consists of 4 members;

İren Berk Özalp

Nurullah Gürcan

Erdem Alpay

Belkay Sarataş

### **2.1.3 Testing Team**

Testing Team consists of 2 members;

İren Berk Özalp

Nurullah Gürcan

### **2.1.4 Release Controlling Team**

Release Controlling Team consists of 2 members;

Erdem Alpay

Belkay Sarataş

## **2.2 Responsibilities**

SDT consists of four members, and this team's main aim is to create necessary modules in our project. We all want to participate this coding part which is so vital for this project, so team is consist of all project participants. SDT is also responsible for documenting releases.

CMT consists of again all four members of our project. This team is responsible for leading team members to obey CMP. Team is also responsible for updating CMP according to our works. Coordination among members of project and auditioning will be handled by this team. So we decide all project members are needed to be present in this team.

Testing Team consists of two members (İren Berk Özalp and Nurullah Gürcan) and is responsible for making necessary experiments on the

works that SDT developed. For our project, this is vital, because our product will have lots of affairs with user inputs. So, in order to avoid bugs, we need to conduct testing carefully with professionally defined inputs.

RCT team consists of two members (Erdem Alpay and Belkay Sarataş) and is responsible for deciding and finalizing the releases of software. They are also responsible for monitoring these releases.

### 2.3 Tools and Infrastructure

In order to achieve coordination all among the group members, we need to use some tools. The main tool we plan to use is SVN. " **Subversion (SVN)** is a version control system initiated in 2000 by CollabNet Inc. It is used to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS)." (en.wikipedia.org)

It is free software and it can be used under our departmental accounts.

## **3. The CM Process**

### **3.1 Identification**

We are going to give information about code and documentation part, which will be identification part of the status of the project.

#### **3.1.1 Code Modules**

We have three different coding modules, which can work independently. These modules have different directories in SVN system. One of them is helper module; this module will have codes about helping functions that we are going to implement in our project. Some of them are parfor, parvecf and operation overloading.

Another module is aiding module, which will include aiding functions of our project. Some parts of this module are script and cluster connection parts. You can find detailed information in our detailed design report.

Last module is proof of concept module, which will include an example of parallelized library for Octave.

#### **3.1.2 Documentation**

Documentation is necessary in big software projects, because without documentation your project can not go through next phases. Sometimes good separation of jobs does not sufficient enough to produce a project, so with the help of documentation job separation can be reassigned

more efficiently. Also documentation makes group members keep in touch and work in a coordinated way.

We are going to publish our;

- Project Proposal
- Requirement Analysis Report
- Initial Design Report
- Detailed Design Report
- Living Schedule
- Configuration Management Plan

under our document section in SVN.

## **3.2 Management and Control**

### **3.2.1 Change Requests**

When a member finds a bug in current version of the project, he must make a change request. This is a vital part of the configuration management, because in well disciplined and well organized projects, changes in current system are not done individually; related members must be informed. In our project every member can make a change request and every member is included in the change request process.

Whenever a member wants to make a change request, he must inform all other members via e-mail, which will have a predefined format. This e-mail must include this information;

- Name of the request maker
- Date and time of the request
- Name of the module and file that will affect
- Description of the request
- Priority of request

That information must be obtained before meeting between team members. Those meetings are going to be held in predefined time and mostly in internet platform. But before these meetings request maker must inspect effects of his request, so that in meetings he can share his information and team can make a decision with the leading of data provided before request maker.

Whenever a change request is accepted by team members, request is applied by the request maker and versioning account will keep for future works.

### **3.2 Configuration Status Accounting**

Process of the project must be accounted and analyzable. Otherwise, we cannot benefit from our department aids for our project. So the process of project will be informed to our assistant in weekly meetings individually. In these meetings, we inform about our weekly process. Also, in the second semester we use SVN tool. With this tool our advisors and assistants can watch our progress in our SVN accounts.

Related documents are going to be provided for them through SVN. Also, we will have a living schedule in our website to show our progress.

### **3.3 Auditing**

Auditing is an important part of the configuration management which will increase the reliability and quality of the product. Mainly audition will be implemented by a member request. We will implement two types of auditing; one is functional, other is physical. Also mostly auditions are implemented before releases, snapshots and finalization of modules and these auditions are planned to be documented for later inspections

### **3.4 Time Plan**

The time plan of our project has been done in the following Gantt chart. This chart will be updated every week throughout the development process according to the changes in our implementation.

ID	Task Name	Start	Finish	Duration	2009																									
					Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep														
1	Analysis & Design	01.01.2009	14.01.2009	2w	4.1	11.1	18.1	25.1	1.2	8.2	15.2	22.2	1.3	8.3	15.3	22.3	29.3	5.4	12.4	19.4	26.4	3.5	10.5	17.5	24.5	31.5	7.6	14.6	21.6	28.6
2	Implementation	15.01.2009	27.05.2009	9w	Group																									
3	Initial creation of parallel package	15.01.2009	28.01.2009	2w	Group																									
4	Creation of Package	15.01.2009	21.01.2009	1w	Group																									
5	ParPrint Implementation	22.01.2009	26.01.2009	1w	Erdem, Berk																									
6	CoreScript Implementation	22.01.2009	28.01.2009	1w	Belkay, Nurullah																									
7	Initial Documentation of package	22.01.2009	28.01.2009	1w	Belkay																									
8	Designing website	23.02.2009	02.03.2009	12w	Group																									
9	Extension of the parallel package	03.03.2009	24.04.2009	7.8w	Group																									
10	Operation Overloading	03.03.2009	23.03.2009	3w	Nurullah, Berk																									
11	ParVecf Implementation	03.03.2009	16.03.2009	2w	Belkay, Erdem																									
12	ParFor Implementation	19.03.2009	08.04.2009	3w	Belkay Erdem																									
13	Gluser Connector Implementation	24.03.2009	24.04.2009	4.8w	Nurullah, Berk																									
14	Testing & Debugging	09.04.2009	24.04.2009	2.4w	Belkay, Erdem																									
15	Documentation of Package	27.04.2009	01.05.2009	1w	Berk																									
16	Parallelizing Linear Algebra Library	27.04.2009	27.05.2009	4.6w	Group																									
17	Implementation	27.04.2009	20.05.2009	3.6w	Group																									
18	Testing & Debugging	21.05.2009	27.05.2009	1w	Group																									
19	General Testing & Debugging	28.05.2009	10.06.2009	2w	Group																									