

Middle East Technical University
Computer Engineering Department

CENG 491

CONFIGURATION MANAGEMENT PLAN REPORT

By
redCat
2007



1321785	Mechmet Kemikli İsmail
1408723	Ruslan Shavaliyev
1462704	Ebru Kuloğlu
1395227	Dicle Berfin Köse
1300862	Gökhan Seyrankaya

TABLE OF CONTENTS

TABLE OF CONTENTS	i
1.INTRODUCTION	2
1.1Purpose of CMP	2
1.2.Scope of Document	2
1.3.Acronyms and Abbreviations.....	3
1.4.References (Related Documents)	3
2.The Organizations CM Framework	3
2.1.Team Structure	3
2.2.Responsibilities	4
2.3.Tools and Infrastructure	4
3.The CM Process	5
3.1.Identification	5
3.1.1.Code:	5
3.1.2.Documentation:	5
3.1.3.Baselines:	5
3.2.Management and Control	6
3.2.1Development	6
3.2.2Deployment	6
3.2.3Change Request.....	6
3.2.4Defect Tracking.....	7
3.3.Configuration Status Accounting	7
3.3.1Project Media Storage and version control	7
3.3.2Reports	8
4.Project Schedules - CM Milestones	9
5.Project Resources	9
6.Plan Optimization	10

1. INTRODUCTION

1.1 Purpose of CMP

The main purpose of this Configuration Management (CM) Plan is to establish and maintain the integrity of the REDDYPOST. The CM plan identify the changes during development process of the REDDYPOST and this is an inevitable in order to sustain control over the whole life cycle of the project.

Software development process is generally achieved by the group of people. In other words, team work is required parallel development. All the implementation and configuration management of the project and description the processes and procedures are defined in this Configuration Management Plan.

1.2. Scope of Document

The most important configuration management activities can be defined:

- **Define and Control**
 - **Responsibilities**
 - **Changes**
 - **Schedules and Procedures**
 - **Version**
- **Test and Verify**

In order to maintain succes, all the members of the project group will have to take time to report the changes defining its requirements and notify the others about changes.

1.3. Acronyms and Abbreviations

CVS	Version Control System
CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
SCM	Software Configuration Management
CRB	Configuration Request Board
SCR	System Change Request
GUI	Graphical User Interface
FPGA	Field Programmable Gate-Array
VHDL	Vhsic Hardware Description Language

1.4. References (Related Documents)

- Software Requirement Analysis
- Final Design Report
- Tutorials of the Hardware products and other technologies used in project

2. The Organizations CM Framework

2.1. Team Structure

Our project group redCat consists of five people. All of the team members are well qualified in programming and software design and moreover have nearly the same experience in hardware Design. So, besides dealing with the design of the phases, all of the project members will be working in the implementation of the project. As a result, the best model fitting our team structure is decentralized management type. However, in order to prevent possible conflicts and future coordination problems we decided to choose a team leader, implying our team structure to be Controlled Decentralized.

In this model, our team will be self organizing. Decisions about the problems and the possible solutions to these problems will be made by group members in weekly meetings. This semester, we planned to have two scheduled meetings in a week.

In our group, nobody has special dominance over the other except from the team leader.

2.2. Responsibilities

As we have a Controlled Decentralized team management structure the following responsibilities are valid for each member of the group.

- To comment on the code properly and explain the changes done efficiently while committing to CVS
- To comply with the CM schedule
- Depending on the CM process, to check in and check out by CVS properly
- Inform the people about System Change Request (SCR) by sending a proper e-mail to the predetermined mail address of the group or by face-to-face communication in the meetings by giving a logical reason for doing it.

The team leader has some more responsibilities which are:

- Coordinating the group members and checking the group members' tasks in orders to realize the critical milestones.
- Assigning tasks to the group members' having light work to finish the unrealized milestones.

The responsibilities of each group member can be reached from the living schedule of the project Digipost.

2.3. Tools and Infrastructure

During the implementation phase of this project, we plan to work in Windows platform. For simple GUI applications, we will get the help of programming language C++ and OpenGL library. For programming FPGA board, VHDL since it has more advantages over Verilog. Moreover, since we have a microcontroller, **MicroC (tentative)** will be used. In order to keep track of project source codes and the versions of them, CVS server will be used.

As it is known, CVS helps the group members work on the same project efficiently at the same. It keeps a central repository that includes source code, past versions of the system, and logs document changes to the system. Team members may check out a directory containing source

code, check in new versions of files, and view logs to learn the modification history of a particular file or directory.

3. The CM Process

3.1. Identification

In order to determine the current state of ReddyPost project, we have identified the following CI's that are subject to change during our project:

3.1.1. Code:

This is the most dynamic part of the project. Thus, we have divided coding part into three modules. These are:

- Administrator Software
- Microcontroller Circuit Software
- Microcontroller FPGA Software

In each module we have assigned 2 main responsible people and one other subsidiary person who will be helping the others in need. Since ReddyPost is a hardware project, it is not very possible to code and test our codes independent from each other. Therefore, these people will mainly be helping each other out in coding.

3.1.2. Documentation:

During the implementation of ReddyPost, the group members will be preparing and changing several documents. These are:

- Living Schedule
- Configuration Management Plan
- Testing Specification Sheet
- Users' Manual.

3.1.3. Baselines:

These are the milestones of ReddyPost. They will be very impelling for us in the implementation part of the project. Some of these milestones are:

- First development and snapshot demonstration

- First release and test specifications
- Presentation
- Final Demonstration

3.2. Management and Control

3.2.1 Development

In ReddyPost project, all tasks distributed to redCat team members according to their skills and interests in subjects. Mainly no one task is going to be done by one team member. Since our team desires to have equal work done by each team member and in the same time to have each task to be done in time, all tasks are distributed in such a way, that at least 2 or 3 team members are being involved in the development process of each task. We think that it's crucial important to manage the project in this way.

3.2.2 Deployment

Deployability is a key **success factor** for any business solution. Any new business solution must fit to the user needs and be efficiently deployed. As a team involved in the development process of the new business solution, we consider the problem of the deployment of our project. Several testing tasks of our project will help us to understand how we can meet the customers' needs and how to make our project to be efficiently deployed.

3.2.3 Change Request

Change Request is considered for the parts of the project which marked as finished. Change Requests raised by any member of team will be discussed in our team meetings arranged twice in a week. Procedure for Change Request is as will be mentioned below:

- Use team's mail account to announce about any Change Request.
- The request should clearly define the nature of the requested change, any deficiencies that the change is expected to correct, problems or symptoms to be addressed, the conditions for satisfaction of the request.

- Change requests failing to meet the minimum criteria for submission may be honored or rejected by the team members.

After requested change approved by the team members, assigning of the change owner is done by the team members. If the requested change need to be done in a short period, then all the team is going to participate in change completing.

3.2.4 Defect Tracking

Defect tracking is the process of finding defects in a product. In our project defect tracking will be performed after each task is done by testing of that task. First of all, the task will be tested on meeting all the customers' needs. Since each task is assigned to some of group members, it will be crucial important that the same task will be tested by the same group members who were assigned to that task. In the case of unsuccessful result of the test analysis, the team meeting will be arranged; the defects of the task and some possible solutions on the defects will be discussed and finally, some team members will be assigned to handle the same task out of the defects met during the test analysis.

3.3. Configuration Status Accounting

3.3.1 Project Media Storage and version control

As mentioned above, we work on a hardware-based project, thus it is more convenient for us to release our changes when working on our FPGA board all together, on Mondays or Fridays.

We are also provided CVS tool by our department, thusly we will be using its repository for project storage, and also have our back ups on our individual portable computers. The procedure followed in CVS is as known:

1. Login
2. Check out
3. Update
4. Commit

After all version control would be handled by our revision control tool CVS. Additionally simultaneous changes are also handled by CVS manager program.

3.3.2 Reports

Communication is guaranteed via our redcatceng@gmail.com e-mail, and five of us own msn messenger accounts. Also we scheduled two to three-hour-meetings on Mondays and Fridays, thus we are managing our time and catch up with each other's work easily.

If we manage to deliver all the minor changes via emails sent to the team, the only responsibility we need to bare would be frequent checks on our e-mail account. But we will bare in mind that very minor changes are not to be released and committed to CVS until they become too serious for the development of the project or some other very minor changes are done.

On the other hand, if we were working on a completely software project we would be using Doxygen for software documentation, but in this situation, explanations within our code will be enough via keeping in mind the facts that there will exist little coding compared to software projects and we have a small group of people.

Additionally, reports are the "quality manager's" of the projects, and important during the project lifecycle. Since we will be in close communication, Change Requests and Builds will not be reported via formal reports but be handled via casual e-mails with brief descriptions. On the other hand, Defect Reports will be taken into account more seriously, since there will be consequences where one will leave a bug to the other project member for a different point of view, so detailed descriptions of defects and their time-based tracking will be handled via reports. Its lifecycle and time of fix will be noted deliberately on the report.

Lastly, project status will be checked from the living schedule and questioned within the team informally since we have well-planned meetings, and good communication via msn messenger, so we will not strictly follow formal procedures for our configuration status accounting for ReddyPost.

4. Project Schedules - CM Milestones

At the beginning of the semester, we have made a living schedule indicating the deadlines of the tasks of the project. The schedule is on the website and is updated regularly considering the time constraints and remaining work. All members have to follow the regulations written in this living schedule. Besides these responsibilities of the team members, group has to make meetings among each other twice a week - Mondays and Fridays and with the instructor once a week - Tuesdays and should report the weekly process. In the meetings among the group members we will analyze current situation of the project, discuss the necessary changes, try to find solutions to several problems, and progress accordingly. In the meetings with our supervisor we will be reviewing the finished part of the project and will take advises of the supervisor about the finished and ongoing parts of the project. Moreover, there are CM milestones of the project according to which we will arrange our progress and implementation of the project. These milestones are the followings with their deadlines:

- First development and snapshot demo 18th of March
- Pic Programming 29th of March
- FPGA Programming 2nd of April
- Testing and Debugging 1st of May
- First release and test specifications 6th of May
- Team presentations 7th – 21st of May
- Final package + Final demo/competition 11th of June

5. Project Resources

In order to control CM activities, we need personnel, tools, training and techniques. Our personnel are the members of the group, the most important resources of our project and are responsible of development and management of the project. The main tool we will be using for CM is CVS. This tool will let us keep the completed files and manage the old and new versions of our codes. Of course, we will need training on how CVS is used and have an understanding on the usage of it: Every member will comment about his work to show the changes he has done. In addition to these sources, we will be using our web page for CM since it includes all the project

documents and living schedule which is being updated continuously. All of these are important resources to complete the project without any problems.

6. Plan Optimization

Since we are a group of 5 people, it is easy to follow the changes and CMP. During the weekly meetings we will analyze the currency of our project and according to our observations; we will update the CMP weekly if it is necessary. In addition, the mail traffic among the members will guide them about the changes in the project. This way, we will prevent CMP from being redundant and outdated.