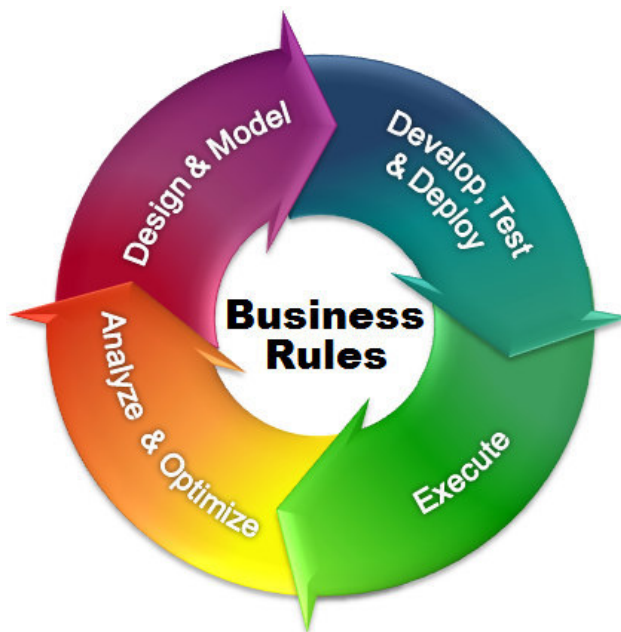


2009

CEng 492

DSK4BRMS

Configuration Management Plan



Yetkin KARIŞ

Metin BARIŞ

Erkan AKYOL

Ghassan ALSHANA

MOCKWARE

3/9/2009

1. INTRODUCTION	3
1.1. The Purpose of the Document	3
1.2. Scope and Overview	3
1.3. Definitions, Acronyms and Abbreviations	3
1.4. Document References	4
1.5. Document Overview	4
2. CM FRAMEWORK ORGANIZATION	5
2.1. Organization	5
2.2. Responsibilities	5
2.3. Tools & Infrastructure	6
3. CONFIGURATION MANAGEMENT PROCESS	7
3.1. Identification	7
3.1.1. Code	7
3.1.2. Data	8
3.1.3. Documentation	8
3.1.4. Baselines	8
3.2. Configuration Management and Control	9
3.2.1. Change Requests	9
3.2.2. Evaluating Changes	9
3.2.3. Implementation of Change Requests	9
3.3. Configuration Status Accounting	10
3.4. Auditing	10
4. PROJECT SCHEDULES & CM MILESTONES	10
5. PROJECT RESOURCES	11
6. PLAN OPTIMIZATION	12

1. INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

Configuration management plan is main part of software development process. The management and maintenance of the software should be achieved with an evolutionary approach. Implementation phase of the project mostly includes updates and reconsiderations of the parts of the software. Since even a minor change in the project can lead major consistency problems, there should be a well-organized configuration management plan. It is the main function of configuration management plan that defining certain procedures to follow in these kinds of problems helps the software protection against change related risks.

1.2 SCOPE OF THE DOCUMENT

This document introduces the configuration management plan of Mockware's senior project. In this part a brief definition of the standards of the project and members' responsibilities are given. This document provides the steps that should be followed when a modification occurs during the development of the software. The intended audience for this document is the members of Mockware, the project assistant and the instructor.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

CI	Configuration Item
CM	Configuration Management
CMP	Configuration Management Plan
CVS	Concurrent Versioning System
CCT	Change Control Team
SDT	Software Development Team
CCB	Configuration Control Board
CR	Change Request

IEEE	Institute of Electrical and Electronics Engineers
STT	Software Testing Team
BR	BRules
CMT	Configuration Management Team
RCT	Release Control Team
SVN	Subversion

1.4 DOCUMENT REFERENCES

- Mockware Requirement Analysis Report
- Mockware Final Design Report
- Software Configuration Management template, METU Computer Engineering – CENG 492
- IEEE Standards for SCMP (IEEE Std 828-1998)

1.5 DOCUMENT OVERVIEW

The main subparts of the document are listed below:

1) Introduction: This part of the document introduces the function and content of the CMP and meanings of the abbreviations and the references used.

2) The Organizations CM Framework: The section describes the organization of the team, the responsibilities of team members and tools used during the project development.

3) The CM Process: This part

- explains how to identify the current state of CIs
- describes the methodologies followed for CM
- defines the status updates to be provided and audit plans

4) Project Schedule – CM Milestones: Time plans are given in this part.

5) Project Resources: The resources needed during CMP are explained in this part.

6) Plan Optimization: This part covers the methods to be followed for optimizing CMP

2. THE ORGANIZATIONS CM FRAMEWORK

2.1 ORGANIZATION

In our group organization, we have chosen yetkin to be the group leader and the configuration manager. He is responsible for tracking changes made to the system and updating the project schedule when necessary. Also, since we are members of 4 persons, all of the team members are considered as a member of Configuration Control Board(CCB).

There are four units controlled by CCB:

- Configuration Management Team
- Software Development Team
- Testing and Debugging Team
- Release Control Team

All of the team members participate to each unit.

2.2 RESPONSIBILITIES

Since all team members participate to all units, we are listing the responsibilities of the units as a single list. All team members are responsible for the below:

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.

Software Development Team (SDT): This team will work for our implementations. Team members will integrate the accepted change request into the required module.

Software Testing Team(STT): 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 STT.

Release Control Team (RCT): This team will monitor the progress of the releases.

Also, we have common CM responsibilities in addition to upper responsibilities. These responsibilities are:

- ✓ Following updated CM schedule and obeying deadlines.
- ✓ Notifying all members about changes on source codes.
- ✓ Obeying team conventions about commenting and code writing.
- ✓ Committing written code regularly.
- ✓ Informing other team members about the progress.

All of the above activities are supervised by project leader.

2.3 TOOLS AND INFRASTRUCTURE

We will use Concurrent Versions System (CVS) as our version control system. CVS server is supplied by our department for us. CVS is an open-source version control system. CVS keeps a central storage that contains current source code, past versions of the system, and logs that document changes to the system. Any previous version of the software and the documentations of it can be reached easily, by the help of these properties of CVS. In addition, it can be controlled that whether any member of the project made any change on a file and which parts of the file was changed. We will use CVS client of Eclipse. Since, we will use Eclipse during the Project and also Eclipse is a suitable CVS tool which works under both Linux and Windows.

Moreover, the advantage of using CVS is that the history of source files can be recorded in a repository and the development of same project in a group can be achieved

easily. Every developer works in his own directory and CVS merges the work when each developer is done. Every team member has to commit and retrieve codes of the project over CVS.

Also, we need to use SVN (Subversion) system as our code repository. With the services of SVN, each team member can work on the source code in a parallel fashion, and get the latest changes made by other members in an incremental way. With SVN, a member can see the updated lines of source code from a creating revision. This feature makes bug tracking easier, because when a new bug is introduced, it is probably result of latest changes to the source code. Also conflict resolution features of SVN makes teamwork on the same source code file very easy.

3. CONFIGURATION MANAGEMENT PROCESS

3.1 IDENTIFICATION

This section covers identification of current states of BR. BRules has CIs which are also covered in this section. CIs are grouped into four main sections which are code, data, documentation and baselines.

3.1.1 Code

The main part of the project is the source code of BRules. Since the implementation of the project takes a long term, there will be some testings, errors or new attributes,so the source code will need changes during the implementatin of the project.The modules of BRules are

- Execution Module
- Management Module
- Connector Module

These three modules are the CIs of our project. All classes will be written in Java and SVN will be used for a coordinated work. With the help of SVN repository the members of team, our assistants and instructors will see the development of the project.

3.1.2 Data

This part will include the source codes of GUI of BRules. For a better visualization, there will be images, icons, buttons etc. These tools will change according to user of the program.

3.1.3 Documentation

For a good software product, well documentation is so important. Being aware of this, we, Mockware team, published following documents :

- Proposal Report
- Requirement Analysis Report
- Initial Design Report
- Final Design Report
- Configuration Management Plan

These documents are accessible at Mockware Group's official web site.

3.1.4 Baselines

Baselines mainly consist of milestones, tasks at distinct stages of the project. We prepared gantt charts to identify the schedules throughout the project. Well documentation is done up to now. In the second term each stage will be controlled by SVN, we will begin the implementation part and track it in SVN. The baselines of the project are:

- Requirement Analysis
- Initial Design
- Final Design
- Configuration Management Plan
- Implementation of modules and integration

- First Demonstration
- Testing
- Documentation
- Final Demonstration

3.2 CONFIGURATION MANAGEMENT AND CONTROL

We have completed our design which has modularity maintained in the first semester. We will surely change some parts of our design. In this part of CMP how will the changes be handled in the project progress will be explained.

3.2.1 Change Requests

As Mocware team we used to use a proxy e-mail address to contact with each other and our assistant. We will also have a forum in our web site. CRs will be discussed in this forum or via our proxy e-mail address. Each member will have a version of his work and make CR to other members with necessary information like:

- Owner of CR
- Time of CR
- Description of CR
- Related module of CR
- Deadline of CR
- Level of CR
- Any other comments

3.2.2 Evaluating Changes

When a CR is posted to group mail or in the forum, the SDT and CCT will evaluate it and a meeting will be arranged for all team members. CR will be discussed in this meeting. Then CCT will choose some members among SDT for handling the CR and put a deadline for assignment according to level of CR.

3.2.3 Implementation of Change Requests

If CR is approved by CCT with consultation of all members, the members of SDT which are chosen by CCT will do the implementation and integration with the needs of CR. Afterwards STT will test the new implementation and integration. After being approved by STT, the changes in source code will be updated by SDT in SVN.

3.3 CONFIGURATION STATUS ACCOUNTING

This part of the project is about recording the status of CIs and changes made on them. As the project grows, configuration status accounting becomes more important. Every status of development of the project will be tracked. Each change will be logged with date and name. Changes will be shown in each version of the product. And also before every change current version will be saved for back-up. SVN will show the progress of development to team members and also to our assistant and instructors.

3.4 AUDITING

Configuration Audit is necessary for high quality and meeting requirements in the analysis part. The audit gives the project team the chance to control the effects of changes so that necessary precautions can be taken. So, as a group, we will have an audit program. All team members will participate in audits and discuss ideas about changes. CMT will be responsible to arrange audits. We will have meetings every week for audits. If CMT needs a necessary audit, they can arrange an audit at any suitable time.

Changes will be done only according to CRs. We will evaluate current situation and changes. We will have functional audits about proper functionality of the project, process audits to verify reliability, maintainability and integrity. These audits will help us with our examination of each stage of the project.

4. PROJECT SCHEDULES AND CM MILESTONES

One of the factors that improves the software product quality and enables the product not to pass over the time which is assigned to the project is the time planning of the development progress. At the beginning of the semester, we have prepared the living schedule for this semester implementation and testing process of the project. We have

milestones which are determined by the syllabus of the course besides our modules. We have divided our project into modules and the modules are our milestones during our software production. These are milestones of the project and the deadlines assigned to them:

Id	Task	date
1	CMP submission	09/03/2009
2	Implementation of Management Module	20/03/2009
3	Implementation of Execution Module	29/03/2009
4	Integration of Modules	07/04/2009
5	First testing and Release version 1.0	15/04/2009
6	Implementation of Interaction Module	05/05/2009
7	Implementation of User Interface Module	21/05/2009
8	Finalizing Integration of Modules	29/05/2009
9	Second Testing and Release Version 2.0	07/06/2009
10	Debugging and Improvements on GUI and Modules	10/06/2009
11	Documentation and the Final Release	15/06/2009

5. PROJECT RESOURCES

We will use following resources for our project:

- All documentation done by us so far including CMP
- Mockware official web site
- Our assistant Ms. Selma(also thanks to her assistance last term)
- Mr. Semih from CyberSoft
- SVN
- Trac

Having response on the project,Mockware team members are the most important resources of this project.

5. PLAN OPTIMIZATION

This CMP document will be a guide to Mockware team in the development process of the project BRules. Our progress will be according to this document and in any further change CMP document will be updated.Every week SVN repositories will be updated. We will have meetings at each week so that scheduling and plan optimization will be done frequently.All changes and progress will be evaluated in these meetings. In any unexpected urgency, we will act instantaneously,not waiting for deadlines.

We will keep track of the project individually and discuss with each other.Since the communnication among group members is crucial,we will also meet informally as much as possible.