



MIDDLE EAST TECHNICAL UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

Senior Project
Configuration Management Plan

A Company Manufactures Everything
(ACME)



Table of Contents

- 1. Introduction..... 3
 - 1.1. Purpose of the Configuration Management Plan 3
 - 1.2. Scope of the Document..... 3
 - 1.3. Definitions, Acronyms and Abbreviations 4
 - 1.4. Document References 4
 - 1.5. Document Overview 5
- 2. The Framework of CM 6
 - 2.1. Organization..... 6
 - 2.2. Responsibilities 7
 - 2.3. Tools and Infrastructure..... 7
- 3. The Configuration Management Process..... 8
 - 3.1. Identification 8
 - 3.1.1. Code Identification..... 8
 - 3.1.2. Date Identification 9
 - 3.1.3. Documentation..... 9
 - 3.1.4. Baselines 10
 - 3.2. Management and Control..... 10
 - 3.2.1. Requesting Changes..... 10
 - 3.2.2. Evaluating Changes 11
 - 3.2.3. Approving or Disapproving Changes 11
 - 3.2.4. Implementing Changes 12
 - 3.3. Configuration Status Accounting..... 12
 - 3.4. Configuration Auditing 12
 - 3.4.1. Functional Configuration Audits (FCA):..... 13
 - 3.4.2. Peer Reviews (PR):..... 13
- 4. Project Schedule..... 13
- 5. Project Resources..... 14
- 6. Updating Plan..... 14

1. Introduction

1.1. *Purpose of the Configuration Management Plan*

Changing is one of the unavoidable facts of life. Moreover changing is more important and more crucial when your job is a part of technology. Computer based applications are very changeable applications in their development cycles. In every software project, some parts of the project tend to be changed and/or modified as the time passes and the things get complicated. No one can exactly know whether the work done is permanent or it is object to change. A software group can have very different members who think differently and make their jobs in different way. Therefore when an inconsistency occurs at any phase of a project, team members are gathered and try to fix that inconsistency. This is usually done by changing some parts of a project. Of course, a change to a specific part of a project can affect other parts of it. Thus, these changes should be handled appropriately by team members.

In this document, we covered the Software Configuration Management Plan (SCMP) which is mainly about handling version and change controls. In other words, for a proper software project, the changes and updates should be first identified, then controlled, managed and finally audited and all the members in the project group should be informed about the concerned change or update. This type of configuration management plan is a must for a proper software project.

1.2. *Scope of the Document*

The main scope of this document is to define and apply SCMP to our project CoreAccess to have a product stable and easy to maintain. Under the name of SCMP Report, this document mainly includes identifying and managing software configuration items, handling version controls, auditing the changes and informing all necessary data to group members.

In this document, we first mentioned about the possible organization and responsibilities of the product CoreAccess for the configuration management. We are aware of the importance of stable organization. After that, the tools and infrastructure about controlling the version and change took place in our document. Our Configuration Management Process is described in mainly four subtopics. These are identification, management and control, configuration status accounting and auditing. This part of the project is very crucial. If we can manage to form a proper configuration management plan considering this process, we think we will prevent some possible problems beforehand.

1.3. Definitions, Acronyms and Abbreviations

These are some abbreviations in the SCMP:

ACRONYMS	DEFINITIONS
CCCT	Configuration Control and Evaluation Team
CMT	Configuration Management Team
TDT	Testing and Debugging Team
DT	Developer Team
SCR	System Change Request
CVS	Concurrent Versioning System
CMP	Configuration Management Process
CI	Configuration items
FCA	Functional Configuration Audits
PR	Peer Reviews
PCA	Physical Configuration Audits

1.4. Document References

We benefited from the following references while forming our SCMP:

1. IEEE Standard for Software Configuration Management Plans (IEEE Std 828-1998)
2. 2005 – 2006 CEng Senior Project Groups Documents (<http://senior.ceng.metu.edu.tr/2006>)
3. “Software Configuration Management” presentation prepared by METU Computer Engineering Department for the course Ceng 492.
4. Version Control, <http://www.tortoise cvs.org>

1.5. Document Overview

According to the references we utilized to create this document, we divided our SCMP Report in six parts:

1. Introduction:

At this part, the general purpose and main scope of the document is stated. Moreover, we gave the references and the definitions, acronyms, abbreviations we used throughout the document.

2. The Configuration Management Framework:

The CM activities which will be done through the project are divided among the group members to make each member be aware of their responsibilities. Besides that, the tools and infrastructure are declared clearly.

3. The Configuration Management Process:

At this part, the technical issues about CM are mentioned. These are mainly identification, management and audit of CIs.

4. Project Schedule – Configuration Management Milestones:

The schedule of CM activities and the milestones for them is located to this part.

5. Project Resources:

The necessary resources for CM activities are stated at this part.

6. Plan Maintenance:

At this part, the CM activities to make the project stable and to maintain the bugs of the project are discussed.

2. The Framework of CM

2.1. *Organization*

CM activities are handled by the following teams, in our project LADES;

- Configuration Change Control Team (CCCT)

This team is responsible from the checking, approving or rejecting the SCR according to the demand from customers or testers. So this team reviews the change request, decide on the consequences of the changes, and controlling the change progress if it is done correctly or not.

- Configuration Management Team (CMT)

This team emphasize on the creation and maintaining the configuration management plan. Also this team is responsible from coordinating and reporting the CM activities during the project.

- Developer Team (DT)

This team is in charge of the development phase of our project. Namely this team is doing the coding part so implements the CM activities.

- Testing and Debugging Team (TDT)

This team is responsible from the testing and debugging of the coded source. This team repots the bugs of the system. And according to the results this team makes the change requests.

Our team is consisting of only five people. Therefore during the formation of the project, every team member has roles in each team.

2.2. Responsibilities

In our project, every team member is responsible from the completion of the tasks. Sometimes the change in roles is done according to the weekly meeting among group members and also according to the feedbacks that we get from our assistant.

Beside these, every team member is responsible from informing other members when a change is occurred on the CVS system and writing relevant comment to the CVS. Also each team member has to give emphasize on completing assigned tasks until deadlines.

2.3. Tools and Infrastructure

The following tools will be used during our project.

- CVS

CVS is the one of the most recently used version control systems and it is preferred as it is open source. CVS support is provided by our department The CVS tool will be helpful for our project in many ways. Namely it simplifies the version control system. And there will be no more conflicts on combining the code modules that are implemented by different team members.

- Eclipse

Eclipse is the tool that is generally used by Java programmers. As we implement our project by Java as ACME we use Eclipse. Eclipse is also preferable as it helps with documentation formation system.

3. The Configuration Management Process

3.1. *Identification*

In order to identify the LADES product of ACME project, the CIs of the project is mentioned. In this project, our documented physical and functional characteristics of the code, specifications, design, and data elements are identified differently according to their responsibility or the parts which they are related with. We have mainly three parts in project: Source Code, Documentation and Data. All identification issues will be related with these parts.

3.1.1. *Code Identification*

Code CI consists of all implemented modules of the project. These modules are separated into twelve parts:

- acme.lades
- acme.lades.database
- acme.lades.property
- acme.lades.property.category
- acme.lades.ui
- acme.lades.ui.action
- acme.lades.ui.menu
- acme.lades.ui.panel
- acme.lades.ui.table
- acme.lades.ui.toolbar
- acme.lades.ui.tree
- acme.lades.xml

All these parts include source codes files related with its name, which increases modularity of the source code files and results in low coupling. Therefore, handling of identifying changes and associating them with the affected CIs are easy. Furthermore naming of these source codes are associated according to the package name that they are related with for assigning unique identifiers to each item to be controlled.

3.1.2.Date Identification

Data CI consists of the necessary components and project settings files which are mostly stable and input to our project. They are mainly identified under six topics.

- components
- events
- image
- layout
- lib
- build

All the related files stored under the directory named according to the data classification, and files beneath them are named according to the path of the directory.

3.1.3.Documentation

In order to clarify the project mechanism, we have to document every phase of the project.

These documents are as followings:

- Project Proposal
- Requirement Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan
- Test Specifications
- Weekly Progress Reports
- Living Schedule

All the related files stored under the directory named according to the data classification, and files beneath them are named according to the path of the directory concatenate with the increasing sequence number such as Weekly Progress Report 1 and Weekly Progress Report 2 and the biggest sequence number implies latest document.

3.1.4. Baselines

Baselines CIs mention the milestones of the project. These CIs are as the followings:

- Analysis of Requirements
- Detailed Design
- Developing Prototype
- SCMP
- Implementation
- Testing
- Documentation
- User Manual
- Installation Plan
- Executables

Controlling status of the all CIs is also very essential. In this respect, CVS is also used throughout the project life cycle. Every CI will receive a new version number when there will be a change in it. Each previous version will be stored in the same CI directory. This issue will also relate the source code naming; the name includes the data in addition to the version number. Moreover, CVS will also construct a library system for all CI items by providing ways to how to retrieve and reproduce controlled items from library storage. Crash recovery tools will also used on order to provide disaster prevention and recovery procedures. By these methods we are able to identify the current state of our overall products and systems.

3.2. Management and Control

In this part of the CMP, configuration control activities of the CA request, evaluate, approve or disapprove, and implement changes to baselines CIs. These activities are explained in following:

3.2.1. Requesting Changes

If a change and modification is necessary for a module, we can request one of the team members to fix it. We accept only our team members and our instructors change request

during implementation. For request and to inform the other team members about request we use our ACME mailing list, for a change request a mail is sent and process goes on the mail named with request id. In SCRs, we will use a template for mailing. This template consists of:

- Id number of the SCR
- Date of SCR
- Deadline of SCR
- Priority of SCR (1 to 4, 1 is highest priority, 4 is lowest priority)
- Owner of SCR
- Assigned member of SCR
- Description of SCR
- Change requested Module

3.2.2. Evaluating Changes

Each member is responsible for (specified in living schedules of the project) determining the affects modules when s/he changes something in his/her module. Furthermore, because of coupling the implementer of a module could require a change in another module. Therefore, team members have to know the function all modules of the project even if they do not know the exact content of the modules. Moreover, when a SCR is created, the priority should be determined according to the milestone. As it is mentioned before, there are 4 types of priority. 1 is the highest priority and 4 is lowest one. If these rules are obeyed, it is easy to fix the SCRs by collaboration between team members.

3.2.3. Approving or Disapproving Changes

The SCRs are approved or disapproved by the CCB. Since all members of the team are CCB, mailing technique is used for this CCA. If one of the members hesitates the effect SCR, s/he can express his/her complain. After discussion, the CCB members decide to approve the SCR or not.

3.2.4. Implementing Changes

After the SCR is accepted, the task assigned person is determined by the SCR creator as it is mentioned before. Then, the responsible person fixes the code according to the description and uploads it to the CVS. Thus, the others member can achieve the latest updated version of the module.

3.3. Configuration Status Accounting

Configuration Status Accounting mainly includes the activities for recording and reporting the CIs of the project. By CSA, we will report to all the group members about the status of the project. In addition to that, our assistant and our teachers will be informed. The related changes used to be informed by meeting reports and weekly meetings at the first semester. At this term, besides meeting reports and weekly meetings, we will use CVS which will include the latest changes and our website to manage the informing business. When a member puts his/her source code to CVS, the related document or comments should be provided by him/her. Moreover, we still carry on the communication by the email among group members and our assistant and our project Google group. When a change recorded, we need some information which must be present for every change. First of all, the description of the change should be stated. This includes the information about why the change is needed and how the change is done. Moreover, we have to observe how this change affects the whole project and check whether there can be an inconsistency or not. Next, the people responsible for the change should be kept. When the things go wrong, the member who made the change knows much anybody else in the group. Finally, the changes must be numbered regularly as versions and the dates must be recorded.

3.4. Configuration Auditing

We will divide configuration auditing process into two main parts. These are;

- Functional Configuration Audits (FCA)

- Peer Reviews (PR)

3.4.1. Functional Configuration Audits (FCA):

FCA generally determines whether the implementations of each CI item are parallel with its design specifications. In addition, FCA verifies that the proposed functional baselines are consistent with the changes.

3.4.2. Peer Reviews (PR):

Peer Reviews involve a methodical examination of software work products by other people outside the developer team to identify defects and areas where changes are needed. The purpose of Peer Reviews is to remove defects from the software work products early and efficiently. So we give extra credit for peer reviews.

In our project, weekly meetings with our teaching assistant Oral DALAY will take the place as peer reviews, where he will evaluate our changes and comment on them. These weekly meetings as peer reviews will help us to resolve problems on a more objective basis.

4. Project Schedule

The main scheduler for our project is the living schedule we have made in the previous weeks. The project group as a whole will try to meet the time requirements that are written in the living schedule. The schedule will be updated regularly to reflect the finished work. There will be two meetings each week: The project group's internal meeting, the meeting between the group and the assistant to report weekly process. The milestones for our project are:

- Implementation of Component Events - 14.03.2007
- Implementation of Code View to Design View Conversion - 19.03.2007
- Implementation of Database support - 16.03.2007
- Improvements to Graphical User Interface - 26.04.2007

- Implementation of File Tracking System - 30.04.2007
- Testing and Maintenance - 11.06.2007
- Documentation - 05.06.2007

5. Project Resources

In order to maintain a good code structure and haste the coding we are planning to use CVS extensively. CVS is a must for projects which are being developed by several developers. CVS will ensure that we will code in parallels and no one will get in another one's way. The changes on the code are kept and previous versions of the files can be reached easily. We will use the built in CVS client in Eclipse.

Another CM source is our web page which includes all the recent project documents and living schedule that is being updated regularly. These resources will provide us a base to complete this project without any problems.

6. Updating Plan

In a group of five people it is easy to follow the changes in the project and configuration management plan. Everybody is responsible for monitoring and updating the plan. In the weekly meeting the group will decide on changes if necessary. If we all agree that a change is necessary we will reflect that change to the plan. The CVS support will be very useful in this changing process. If an urgent change will be made communication between the group members might be a problem. To avoid this problem we have already formed a mail group and everyone is responsible to check his/her mails regularly.