

MIDDLE EAST TECHNICAL UNIVERSITY



COMPUTER ENGINEERING



LIFDsoft

Configuration Management and Development Plan



Topic: Lighthouse Interactive Form Designer

Teaching Assistant: Umut Erogul

1.0 INTRODUCTION.....	3
1.1 Purpose of the Configuration Management Plan.....	3
1.2 Scope of the Document.....	3
1.3 Definitions, Acronym and Abbreviations.....	3
1.4 References.....	4
2.0 ORGANIZATIONS CM FRAMEWORK	4
2.1 Organization.....	4
2.2 Responsibilities.....	5
2.3 Tools & Infrastructure.....	5
3.0 The CM PROCESS.....	5
3.1 Identification.....	6
3.2 Management and Control	8
3.3 Configuration Status Accounting.....	9
3.4 Configuration Auditing.....	10
4.0 PROJECT SCHUDULE.....	11
5.0 PROJECT RESOURCES.....	12
6.0 PLAN OPTIMIZATION.....	12

1. Introduction

1.1 Purpose of the Configuration Management Plan

We will be faced with lots of changing situations in development period. The more change encountered means more complicated things we are going to deal with in future. Because many tasks, codes are going to be modified, there exists inconsistency danger in project features. We need a mechanism that helps us in handling confusion because of updates. The purpose of the Software Configuration Management (SCM) is keeping the general control and continuity of the project. In computer based project, SCM organize changes and updates in development period.

1.2 Scope of the Document

We introduce our SCM plan and how we carry it on project in this document. We start with organizations, and then pass to responsibilities for the configuration management and finally the tools and infrastructure used for configuration management. Thereafter, we will talk about the configuration management process in four topics which are:

- Identification
- Management and control
- Configuration status accounting
- Auditing

We will describe the project schedule and configuration management milestones, project resources and plan optimization at the end.

1.3 Definitions, Acronym and Abbreviations

CI	Configuration Item
CM	Configuration Management
CCB	Configuration Control Board
CVS	Concurrent Versioning System

GUI	Graphical User Interface
SCM	Software Configuration Management
SCR	Software Change Request
XML	Extensible Markup Language

1.4 References

We benefited from the following references to form our SCMP:

- IEEE Std 828-1998 IEEE Standard for Software Configuration Management Plans
- Configuration Management presentation prepared by Middle East Technical (METU)
- 2006-2007 Computer Engineering Senior Project Groups Documents.

2. Organizations CM Framework

We determine the organization of roles due to the needs of software. We deliver the roles and responsibility of given a role is permanent if any problem does not occur. We explain organization in part 2.1 in detail.

2.1 Organization

All of us are part of CCB. The main characters of CCB are:

- Control of CVS
- Signal mechanism for members to realize change in CVS
- Adding comments to code in CVS
- Controlling CVS performance of members
- Setting up baselines.
- Applying of CM activities.
- Performing CM policies.
- Completing the tasks in determined time.
- Controlling schedule and plans.
- Identifying the configuration items.

2.2 Responsibilities

In our project, team members are concentrated on their own tasks. The tasks are usually personal but some changes may occur with respect to meeting results in the week. We are planning to establish a quality control team to guarantee of convenience between CMP and development process. It will also provide to control works of each. We have CM Update team for updating the CM schedule due to completed and incomplete tasks. It will report CM activities. We must ensure that the control of change is done properly. Therefore, we have Testing team for testing the results of modifications and reporting errors if exist. Finally, we have Development team for generating baselines and deal with changes in development process. It will also executes and have some try on our form designer to get feedbacks.

2.3 Tools & Infrastructure

We advised to use CVS as version control system. Thanks to our department for providing CVS server to us. We will benefit from CVS in different ways. It saves a central storage which includes current source code, pre versions of the system and logs of changes. It gives incredible fallowing occasions to members about handling the changes made in code. We can access to initial versions of any file. It is important to control that whether any change is made or not. Besides, CVS prevents conflicts on merging the modules implemented by separate team members. We use the CVS by the help of Eclipse. It has user interface option for connecting to CVS. Moreover, with the help of Eclipse we can work both in Windows and Linux.

3. The CM Process

The configuration management process helps us to manage the configuration of the software. There must be some pre-defined managerial and technical SCM activities to administrate CM. These activities are divided into four main groups:

3.1 Identification

CIs (Configuration Items) are important for our project to identify our current state in LIFD. We have chosen following items as our CI, because they are the most important parts of our project. They can change every time and LIFD will be affected by these changes, and all identification issues will be related with these parts:

- Software
- Database Objects
- Data Files
- Baselines

3.1.1 Software

Software is main part of our project which consists of four main modules (GUI, XML, Database and File Operations). Each will contain numerous source files. These modules are defined in design stage by grouping similar functionalities together. We may add some new modules if there is a need to add new functionalities. However, we can combine two modules into one module to implement effectively. As a result, coding is the most changeable part of LIFD. We must obey some pre-defined standards to generate new understandable codes as we mentioned in Requirement Report. Some of them are:

- Variable names must be meaningful.
- Function names must be meaningful and should reflect responsibility of function.
- If variable name have more than one word, all word but not first must begin a capital letter.
- Comments must be placed before functions and at the beginning of the class.
- Class names must be meaningful.

3.1.2 Database Objects

Database objects are XML codes created to store necessary information of form to help loading and saving functionality. This information is kept in hibernate database server on a table.

3.1.3 Data Files

Data CI consists of the necessary components and project settings files which are mostly stable and input to our project. They can be divided into five groups:

- Components
- Events
- Layout
- Lib
- Build

3.1.4 Documents

Detailed information of LIFD should be documented as a necessity for the software management. It helps to clarify project mechanism. Up to now we have written these documents:

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

Our documents' name is in lower case and all words in a document name are separated by underscore (_). An example is configuration_management_plan.doc.

3.1.5 Baselines

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

- Report Deadlines
- Prototype Demo
- SCMP
- Implementation
- Testing
- Documentation
- Installation Package

These baselines will be used as checkpoints of the development process. CVS server will be used to physically define these baselines.

3.2 Management and Control

Although we have done task assignment of modules for the implementation, assignment of one team member may affect other team member assignment when we look at the structure of our project. 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 member fixes 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 department mailing list, for a change request a mail is sent and process goes on the mail named with request id.

3.2.2 Evaluating Changes

Each member is responsible for determining the affected 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.

3.2.3 Approving or Disapproving Changes

With the information given by the change requester and if the change affects other modules we arrange a meeting for discussing the approval of this change. If the change is local it's enough to have approval of each member via e-mail. If not; in meetings, we should pay attention:

- An estimation of implementation time deviation from the previously constructed schedule will be prepared.
- Technical availability of the change will be discussed. Project manager will evaluate the change in the overall project progress aspect. Then, a decision is made by project manager.

3.2.4 Implementing Changes

After the SCR is accepted, the task assigned person is determined by the SCR creator. 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 about the status of the project to all the group members. 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. When a member puts his/her source code to CVS, the related

document or comments should be provided by him/her. Moreover, the status of the project will be seen in Living Schedule added to our web site. In case of a change, a report should be generated including; information about changes, revisions, name of the files subjected to change, reason for changes and the name of the group member who wants modification. These reports can be accessible from our website. These reports will be merged to trace process and backtrack in case of a bug.

3.4 Configuration Auditing

Software Configuration Audit is one of essential parts of our project. It basically contains several independent checking processes for the newly integrated functionalities and modified project components. It aims to preserve completeness and consistency of project modules. After every modification, auditing stage takes place regularly. The group members continue auditing until the configuration status is capable as we declared in our reports. We will control the changes for being sure about their effectiveness and accuracy for our project functionalities. By audits, the group members understand the differences between baseline configuration and current configuration and make decisions about the necessities of the status of configuration. We are going to follow an auditing process consisting of Functional Audits, Process Audits and Peer Reviews.

3.4.1 Functional Configuration Audits

It aims to understand whether SCI is well-formed and the version has all the necessary functionalities or not. FCA generally determines whether the implementations of each CI item are parallel with its design specifications. Before release of the final package, we have to make sure that we have implemented all the requirements.

3.4.2 Peer Reviews

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.

3.4.3 Process Audits

The process audit is concerned with the validity and overall reliability of the development process itself. In this type, the important question is: Is the process consistently producing acceptable results? Since the development process is continuous, we are going to make this kind of audit regularly during the implementation stage.

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.

Team members have meeting with supervisor on Thursdays every week. In these meetings members and supervisor discuss about current level of the project. After this meeting LIFD members make a meeting for coming week, each member gives information about her/his part. There are some milestones for our project which are:

- **First Development Snapshot Demo:** It is the modular version of the first semester's prototype. All the modules will be identified and new modules can be added upon the first semester's prototype. Its deadline is 13.03.2008.
- **First Demo:** It is the milestone of our project which the skeleton must be finished. By this prototype GUI and XML modules will be complete and some progress in other modules are necessary. Its deadline is first week of April.
- **Second Demo:** After this milestone no architectural change can be made in the project. Database module will be completed and significant progress will be done on File Operations module. Its deadline is first week of May.
- **Final Release & Documentation :** It is the end of the project. All modules must be finished till then , and integration, testing and documentation must be completed. Its deadline is second week of June.

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. Without using CVS, there appeared important change inconsistencies in our work in previous semester. Because, it is hard to maintain the code with the changes added by each team members. Therefore, CVS is very important in means of reliable configuration management. All team members will be able to keep up with the changes made on the code. We will use the built in CVS client in Eclipse and CVS which is provided by our department.

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.

Finally, since all team members will be responsible for keeping track of the changes and informing all other members, LIFDsoft members constructs the most important part, human resources part of the project.

6. Plan Optimization

In a group of four 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.