

[TURKUAZ]



MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

‘Text Mining On Turkish Medical Radiology Reports’

**CONFIGURATION MANAGEMENT
PLAN**



SELVİ BOYLUM AL
Yazılım 

Spring, 2007

Esra Zeynep Abacıoğlu – 1394568

Kerem Hadımlı – 1448752

Çiğdem Okuyucu – 1448976

Makbule Gülçin Özsoy – 1395383

İpek Tatlı – 1395557

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1. Introduction	3
1.1 Purpose of CMP.....	3
1.2 Scope of Document	3
1.3 Definitions, Acronyms and Abbreviations	3
1.4 Document References.....	4
1.5 Document Overview	4
2. The Organizations CM Framework	5
2.1 Organization	5
2.2 Responsibilities.....	6
2.3 Tools & Infrastructure	6
2.3.1 SVN	6
2.3.2 Eclipse	7
3. The CM Process.....	7
3.1 Identification.....	7
3.1.1. Source Code.....	7
3.1.2. Data.....	8
3.1.3. Documentation.....	8
3.1.4. Baselines	9
3.2 Management and Control	9
3.2.1 Change Request	9
3.2.2 Evaluating Changes	10
3.2.3 Approving or Disapproving Changes	10
3.2.4 Implementing Changes	10
3.3 Configuration Status Accounting	11
3.4 Auditing	11
4. Project Schedules – CM Milestones	12
4.1 Project Schedule	12
4.2 CM Milestones	12
5. Project Resources	13
6. Plan Optimization.....	13

1. Introduction

1.1 Purpose of CMP

Software projects consist of design, development and implementation details which may be changed or updated during the life cycle of the project. Any change or update in some part of the project affects other parts of the project. Moreover, projects are accomplished by a group of people. Any change or update which is done by a member must be documented and all of the team members must be informed of these changes to avoid any possible inconsistency. Therefore, planning is necessary to handle such changes and updates.

Configuration Management Plan is very important for establishing and maintaining the integrity, consistency of the project throughout the development process. It will also handle versions and change controls. It will also support team members to work together concurrently. Good planning enables an effective communication between the members and eases the process of controlling the development process. All of these reduce risk of the project development.

Our purpose of preparing a Configuration Management Plan is managing the configuration of the “RadioRead” Project throughout its lifecycle, which is conducted by a team of five members. Identification, management, control and auditing of the changes and updates will be decided during the preparation of this document.

1.2 Scope of Document

This document is prepared to supply the relevant policies and procedures for the development of RadioRead. It describes the organization and responsibilities for configuration management of RadioRead and the CM process which includes the identification and management of CIs, version control and change audits. Moreover, we will declare the CM milestones and schedule, project resources and the plan optimization strategies. In other words, we describe all SCM activities and terminologies.

1.3 Definitions, Acronyms and Abbreviations

CA Configuration Auditing

CCB Configuration Control Board

CI Configuration Item

CM Configuration Management
CMP Configuration Management Plan
CVS Concurrent Versioning System
SCM Software Configuration Management
SCMP Software Configuration Management Plan
SCR System Change Request

1.4 Document References

1. The Beginning to your CM Solution

http://www.sei.cmu.edu/legacy/scm/papers/CM_Plans/CMPlans.MasterToC.html

2. IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998

http://standards.ieee.org/reading/ieee/std_public/description/se/1042-1987_desc.html

3. Software Engineering: A Practitioner's Approach – Roger S. Pressman, 5th Edition

4. Software Configuration Management – METU Computer Engineering CENG 492

<http://ceng.metu.edu.tr/courses/ceng490/>

1.5 Document Overview

Our SCMP consists of six main parts, each of which is described briefly below:

- **Introduction:** The introduction part explains the purpose of preparing a CMP and the scope of this document. Moreover, the abbreviations and terms are defined as well as the references which have been used.
- **The Organizations CM Framework:** This part explains the organization and the responsibilities of all group members for the CMP. Moreover, we explained the tools that are going to be used as well.
- **The CM Process:** This part explains the identification, management, and the auditing of the CIs.
- **Project Schedule – CM Milestones:** This part explains the schedule for the CM activities, and we stated the milestones and deadlines.
- **Project Resources:** This part explains the necessary resources that will be used for CM activities.
- **Plan Optimization:** This part explains the methods that can be used to optimize the CMP.

2. The Organizations CM Framework

2.1 Organization

Organizational units should be described properly in order to implement SCM activities. RadioRead has a hierarchical structure for its organizational units. CCB (Configuration Control Board) that handles CM activities is on the top of this hierarchy. All the members of RadioRead work for CCB unit and other organizational units which are in contact with each other. These are:

1. CMT: Configuration Management Team
2. DT: Developing Team
3. TDT: Testing and Debugging Team
4. RCT: Release Control Team

Configuration Control Board (CCB)

This unit reviews SCRs offered by testing and debugging team and analyzes the effects. It also accepts or rejects the SCRs. Audits are held by this unit. Moreover, it reduces the negative effects of changes that have been made.

Configuration Management Team (CMT)

This unit creates, maintains and coordinates CMP and also reports the CM activities to other units during the project. Moreover, CM schedule is updated by this unit according to the works done.

Developing Team (DT)

This unit implements the project source code so the CM activities. Because of the importance of this team, it is in charge of the development phase.

Testing and Debugging Team (TDT)

This unit has the responsibility of testing and debugging the source code. It offers SCR according to the result of testing process.

Release Control Team (RCT)

This unit creates the baselines and controls the releases of the project.

2.2 Responsibilities

We have divided our project into some modules according to their functionality. Each member is responsible from some modules. As mentioned above, we are all in CCB unit; but we have also different roles in the project. The changes are made according to our weekly meetings and the feedback from our teaching assistant. Besides, each member informs others when there is a change on the SVN (will be explained later) system, and writing comments while committing into SVN.

2.3 Tools & Infrastructure

RadioRead uses SVN (instead of CVS) and Eclipse as project resources.

2.3.1 SVN

We use SVN (Subversion) with Tortoise, an open source version control system, as our version control system. Subversion is released under the Apache License, making it free software. It allows the documents to be versioned in a central repository and allows concurrent access to the documents. Besides the current source code, it also includes previous versions of the system. It is a good facility that users can merge the changes and SVN records logs of the changes. And there will be no conflicts when different members changed the source code.

Maintenance of repositories and modules are easier in SVN than in CVS. These are some other reasons why we choose SVN:

- Support for versioned renames/moves (impossible with CVS)
- Supports directories natively: It's possible to remove them, and they are versioned
- Overall revision number makes build versioning and regression testing much easier, as the repository itself has a single revision number at an instant
- Atomic commits
- Intuitive (directory-based) branching and tagging
- Prevents accidental committing of conflicted files

2.3.2 Eclipse

Eclipse is a development platform which is mainly used for developing Java code. Eclipse will be used as the development environment since we will develop RadioRead with Java. Moreover, it works on both Linux and Windows operating systems.

3. The CM Process

3.1 Identification

The CIs of the project are mentioned to identify RadioRead. We identify characteristics of the code, specifications, design and data elements according to parts which they are related with. All identification issues will be related to Source Code, Data, Documentation and Baseline.

3.1.1. Source Code

The main part of the project which is frequently subjected to be changed is the source code. We have divided our project into modules to control modifications of application code. We are using Java as programming language and SVN for concurrent development. All source code will be under its directory in SVN as a good practice of software control. Moreover, we are using package feature of Java for subdivision of source code. It is good for avoiding source mess.



Figure 1: Java Package Hierarchy of RadioRead

3.1.2. Data

Data CI consists of the necessary components and project setting files which stable. *Libraries* and *jar files* will be stored under revision control. Moreover, *information* that we have extracted by text-mining techniques (e.g. Location, Quality, Findings) will be stored. The management of these data in a consistent manner will be controlled by database access module.

3.1.3. Documentation

We have already documented every phase of the project in order to clarify their relationships and interactions systematically for an organized development period. We have partitioned the project into modules and we have considered their interaction in a detailed way for a reliable product at the end of development period.

Documents are as follows:

- Project Proposal
- Requirements Analysis Report
- Initial Design Report
- Final Design Report

- Configuration Management Report
- Weekly Progress Reports
- Living Schedule

3.1.4. Baselines

Milestones of the project are mentioned as baselines CIs. These CIs are the followings:

- Project Proposal
- Requirements Analysis Report
- Initial Design Report
- Final Design Report
- Prototype Demo
- SCMP
- Implementation
- Module interactions
- Testing
- Documentation
- User Manual
- Installation Plan

This list consists of what we have completed and that we plan to accomplish during this semester. Each report and milestones, mentioned in the list, contributed us to consider modules, their interactions, data management and new features carefully.

3.2 Management and Control

Configuration control activities of CA request, evaluate, approve or disapprove, and implement changes to baselines CIs in this part of the CMP.

3.2.1 Change Request

During the implementation of the project changes and modifications of the modules can be requested by the members, instructor, assistants and supporting company. For these requests we use our sbawiki and mailing group. As we are a group of five, team members can

communicate through phones or mailing lists for such a SCR; but for larger projects and larger teams, systematic is as follows:

- Id number of the SCR
- Date of SCR
- Deadline of SCR
- Related Module 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
- On which version change will be applied

3.2.2 Evaluating Changes

Each member is responsible for the changes that he/she made in a module. Since a change in a module may affect other modules, each team member should know the functionality of all modules in the project. When a SCR is requested, the priority should be determined according to our milestone list. As we have mentioned, there are four types of priority (1: highest, 4: lowest). If we consider the methods above, it will be easy to fix the SCRs by collaboration between team members.

3.2.3 Approving or Disapproving Changes

The SCRs are approved or disapproved by CCB which includes all team members, so we use mailing technique for CCA. If any member of the group hesitates on any SCR, all group members meet and discuss. After discussion, the CCB members decide to approve the SCR or not.

3.2.4 Implementing Changes

After SCR is accepted, the one who is responsible of that module will implement the requested change. After the implementation the code is uploaded to SVN by the same member. So, other members can achieve the latest, updated version of the project.

3.3 Configuration Status Accounting

CSA includes activities for recording and reporting the CIs of the project. All the group members, instructor, assistant and supporting company are informed about the status of the project by CSA. The related changes are mentioned every week in meeting reports and weekly meetings. In addition to reports and meetings, we will use SVN, which will include the latest changes. Moreover, our website manages the informing business. When a source code is changed and committed into SVN, it must be commented by the related user. In fact, mailing list and wiki help us to inform these changes. Description of the change which includes the information about why the change is needed and how the change is done should be stated. We have to observe whether the change affects the whole project and controls if there is any inconsistency or not. The member who made the change should be known in a emergency situation like corruption in a module after implementing a change. Finally, the changes must be numbered as versions and dates. SVN automatically does this after each commit.

3.4 Auditing

For a high quality product and satisfied customers configuration management audits are very important. A regular auditing will give us the chance of observing if the changes are done properly or not. Moreover, we will have a chance to see all the members, instructor, assistant and supporting company have information about the changes. We also think that CA's give us information about the progress of the development. For RadioRead we will make main audits before each release and at the same time we will continue to conduct our regular meetings. We will talk about design and implementation problems, what we have done during the week tasks of the next week, change requests and status of the configuration items in these meetings. The first thing that we consider in main audits is to achieve the design goals for the release. All the project members will participate for audits and reviews. The results of these audits will be stored in our SVN repository or our SBWiki.

In our project, every module must meet certain requirements. In order to meet all requirements, we will test each module then we will test interactions between them after implementing each part. The person who is responsible for that module will also be responsible for the audit of it.

After the integration of all modules, physical configuration audit is applied to final version. It verifies that related design decisions matches the all items and confirms that quality control, management, testing, etc. are well planned and applied.

4. Project Schedules – CM Milestones

4.1 Project Schedule

We have prepared a week-by-week plan for RadioRead's implementation. RadioRead project's living schedule can be found at our website. 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.

4.2 CM Milestones

The milestones for RadioRead are:

- Database Schema Review : 17.03.2008
- Implementation of new NounPhraseParser : 17.03.2008
- Implementation of ResultsSectionMiner : 24.03.2008
- Implementation of Database Layer for Miner Module : 24.03.2008
- ExternalWordQueryManager implementation : 24.03.2008
- Implementation of Report Miner : 31.03.2008
- Auditing, reviewing and planning : 31.03.2008
- Implementation of Database Layer for DataEngine module : 07.04.2008
- Implementation of DataEngine module : 07.04.2008
- Implementation of ComplexQueryHandler module : 07.04.2008
- Binding ComplexQueryHandler and DB Layers, finishing ComplexQueryHandler implementation : 14.04.2008
- Implementation of FindingQueryHandler : 14.04.2008
- Design and implementation of Complex Query GUI : 14.04.2008
- Main GUI implementation : 21.04.2008
- Testing and Maintenance - 19.05.2008

- Implementation & Final Demo Completion : 26.05.2008
- Documentation 02.06.2008

5. Project Resources

We use SVN to apply our CMP and follow the CM activities. SVN is a must for projects which are being developed by a team. By using SVN, changes and updates to the source are handled easily and also with the help of versioning feature no old information lost. The changes on the code are kept and release versions of the files can be reached easily. Moreover, our website is another CM including all recent project documents and living schedule which is updated regularly. These resources will provide us to complete the project without big problems.

6. Plan Optimization

In a group of five, it is easy to follow the changes in the project. Everybody is responsible for changing and updating the plan. The group decides on changes and updates in the weekly meetings. If we all agree that a change is necessary, the change will be reflected to the plan. SVN will be very useful in such process. If an urgent change is needed before the weekly meetings, our mailing group will help us to communicate. Each member is responsible for checking his/her mails regularly.