SENIOR PROJECT
SPRING 2008
Configuration Management Plan

MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

Görkem KURT	1395250
Kazım IŞIK	1448760
Okan AKALIN	1448273
Yunus OLGUN	1448992

Table of Contents

1. Introduction

1.1. Purpose of Configuration Management

The main purpose of software development lifecycle process is based on incremental changes on any part of the project such as source codes, releases, and the plans. These incremental changes causes a necessity of the duty of tracking of the differences between the versions of the software. When not tracked, the changes mentioned above are subject to cause unpredictable huge problems. Inevitability of these changes on the project and the possible catastrophic results makes the SCM extremely important. Software Configuration Management aims

-to prevent the situations that can be caused because of the lack of information between software development team.

-to supervise the release of the software product.

-to revise the consistency of the combination of the modules.

-to assist the communication between the members of the software development team.

1.2 Scope of Document

The CMP defines how the members of Okan group will apply CM during the development of Photokan project. The framework of CM, necessary software, details of communication mediums, plans and schedule for the development are given providing a complete discussion of configuration management.

The CMP gives the organization of teams, details the infrastructure, the tools that will be used for CM activities, providing an overview of CM framework organization. The configuration management process is explained. Configuration identification, configuration management and control, status accounting and auditing are discussed. The project schedule is supplied and an overview project resources and plan optimization is given.

1.3 Definitions, Acronyms and Abbreviations

CM Configuration Management

CMP Configuration Management Plan

SCM Software Configuration Management

CCB Configuration Control Board

SDT Software Development Team

TT Testing Team

RCT Release Control Team

CMUT Configuration Management Update Team

1.4 References

1. "Software Engineering a Practitioner's Approach 6th Edition"
Roger S. Pressman, 2001, New York, NY. McGraw-Hill

2. "Detailed Design Report"
Okan group

3. "IEEE Standard for Software Configuration Management Plans"
http://www.ceng.metu.edu.tr/courses/ceng490/documents/private/ieee-std-828-SCMP.pdf

2. CM Framework Organization

2.1 Organization

CM framework organisation of Okan group consists of 5 teams. Every team works on a different aspect of CM. CCB team functions as the supervisor of all CM activities.

2.1.1 Software Development Team

Software development team maintains the Photokan project by implementing the modules and change requests. The members are:

Gorkem Kurt

Kazim Isik

Okan Akalin

Yunus Olgun

SDT makes necessary assignments for

- the implementation of the modules according to the design report
- the implementation of change requests according to the feedback provided by testing team
- maintanance of releases

SDT members regularly update CVS server, providing the means of reviewing and ensuring progress.

2.1.2 Testing Team

Team member: Gorkem KURT

TT determines whether a requirement upon the application is satisfied and whether the individual parts of the system function properly.

Bugs in the application are reported through forms to the configuration control board.

TT member

- creates and applies unit tests against individual parts of the source code
- creates and applies test cases to determine if functional requirements are satisfied.

- reports bugs in the application

2.1.3 Configuration Control Board (CCB)

All team members of Okan group participate in CCB. The CCB works on the basis of consensus between members.

CCB supervises all CM activities.

CCB

- holds and reviews FTRs regularly
- accepts or rejects software change requests suggested by members of SDT and TT
- monitors whether individual steps of development process are in par with the current CMP.

2.1.4 Release Control Team (RCT)

Team member: Kazım IŞIK

RCT member monitors the current release of Photokan.

2.1.5 Configuration Management Update Team (CMUT)

Team member: Yunus OLGUN

CMUT member updates the CMP according to the current state of the project.

2.2 Responsibilities of SCM Members

Responsibilities of SDT members:

SDT members take part in configuration control, reviewing and auditing of code base by:

- proofreading the source code submitted by other members
- coordinating the implementation process according to the schedule

CCB members take part in all CM processes (configuration control, configuration identification, status accounting, reviewing and auditing) as reviewers and validators.

TT members directly perform status accounting and they take part in configuration control as testers.

RCT member reviews and validates the configuration control process.

CMUT member also performs status accounting by comparing the current status of the project against the requirements in the schedule. CMUT member also takes part in the review and validation of configuration control process.

Every member of Okan group follows a common rationale and

- keeps the CVS repository up to date, and informs other members
- through a communication medium.
- logs his progress and comments on the central repository.

2.3 Tools and Infrastructure

Okan group makes use of CVS as its version control system to keep track of changes to the project. The CVS repository keeps the code base, documentation, website, project data along with the change history. Besides its function as a way of reviewing the progress, it also harnesses the power of concurrent development among team members which is vital to the smooth development of the application. CVS also improves the communication among the team members and they can benefit from it to locate the bugs in the software.

The team members will benefit from the TortoiseCVS, the command line CVS client and Eclipse IDE to perform all SCM activities. The necessity of such high number of clients is a result of development performed on multiple platforms. Eclipse is a cross platform IDE and can easily be adapted to use for editing the source code modules. TortoiseCVS works on Windows platform and can be used to perform SCM on all modules. The command line CVS client works on both Linux and Windows platforms and can also be used to perform all SCM activities.

3. Configuration Management Process

3.1 Configuration Identification

As mentioned earlier, a modular design is employed to create a flexible inventory. Photokan project already consists of independent processes so the process of partitioning configuration items into modules is quite simple. The application code is located in directories according to their modules. The project data, test data and documentation also are placed in CVS repository.

Table.1 Modules and their descriptions

Data	This directory contains icons, sounds etc.
DEMExtractor	This directory contains DEMExtactor module.
DEMDisplayer	This directory contains DEMDisplayer module.
Documents	This directory contains the user manual, installation manual.
FileLoadSave	This directory contains FileLoadSave module which is responsible of serialization process.
ImageProcesser	This directory contains ImageProcessor module.
ImageRegisterer	This directory contains ImageRegisterer module.
Mosaicker	This directory contains Mosaicker module.
OrtophotoGenerator	This directory contains Ortophoto module.
ProjectManager	This directory contains ProjectManager module.

TestData	This directory contains test data related with Photokan.

3.2 Management and Control

Management and control is very important for the smooth progression of the project. Every aspect of CM is planned carefully and works according to predefined rules. A systematic approach is taken to avoid communication problems resulting from the absence of proper definitions and rules.

3.2.1 Change Requests

Change requests are one of the most common activities of CM since continous evolving is in the heart of software development. Change requests are planned down to their every detail to fasten the review and decision making process.

3.2.1.1 Change Request Definition

Changes may be requested by TT member in order to remove defects in the software, or to satisfy the requirements of the design. As reviewers of code base, SDT members may also request changes. Team members deliver their requests through a script on department's machines which save the form on an xml database. Change request form consists of the

- name of the member making the request
- name of the files and configuration items along with their version numbers effected by the request
- functional description of the request
- priority of change request
- final decision on request
- the id of assigment form
- date of the request (automatically generated)
- id of the request (automatically generated)

3.2.1.2 Change request evaluation

The change requests are evaluated on weekly meetings by CCB. The design document, snapshots, living schedule and configuration management plan form the basis of baselines for the evaluation of change requests. Since the whole team members are in CCB, necessary assigments can be made in case the request is deemed necessary. Photokan has a highly modular design, and members working on initial revisions of a module can easily take part in implementing changes regarding the request. If a request is approved, an assigment form will be filled, and the xml database will be updated with the form.

The assignment form consists of:

- description of assignment
- team members assigned with the job
- the branching point if necessary
- deadline of assignment
- id of the assigment form (automatically generated)

- date of the assignment (automatically generated)

3.2.1.3 Change Implementation

If the change request is approved, the regarding modules are reviewed during the weekly meeting, and CCB decides whether a branch point or a modification on the structure of CVS repository is necessary. After versioning is finalized, necessary assignments are made and the assigned SDT member makes the necessary changes and commits to the CVS repository.

3.2.2 Deployment Practices

A cronjob which executes a script on department's machine creates daily snapshots which are deployed on the website as a tarball. Besides the daily snapshots, the releases are also deployed on the website. The tarballs employ a make system which install the software on linux machines.

3.2.3 Defect Tracking

TT member is officially responsible for defect tracking but other members are also committed to creating a stable and reliable software. They may and should report the defects detected in the software. The changes to remove defects are requested through forms located on a xml server. As for a defect database, Bugzilla which is a highly regarded defect reporting tool in open source arena will be used.

3.2.4 Development Management and Control

Okan group has established policies to ensure that the outputs of development process works flawless and that every team member can easily adapt himself to the modules that he did not develop. These policies involve:

- Coding Conventions: Okan group has adapted Sun Microsystems Java Coding Conventions. Both Java and C++ share many concepts so this convention easily fits to the C++ scheme. This convention model has proved to be very understandable and is highly regarded among object oriented programming enthusiasts.

- Naming Conventions: The name of the files and modules also follow a convention. This standard facilitates a conscise organisation.

- Code documentation: Okan group uses Doxygen to create the documentation of code base. This facilitates commenting in parallel with coding so that it does not become a burden later. The comment style complies with Doxygen's style.

- Document Description: Every module has a header which includes the name of the contributors, the description of the module and a change history. Okan group follows gnu conventions for document headers.

3.2.5 Engineering Management andControl

As mentioned earlier, CCB holds formal technical reviews regularly. FTRs provide the grounds to evaluate the architecture and to give a chance to developers to reflect their ideas regarding the modules developed by them. Also the whole software development process can be critized and be altered to fit the needs.

3.2.6 Systems Management and Control

The whole process is monitored by CCB and weekly meetings are held to share comments on the flow of the ongoing process. Another important facility is the CVS usage. CVS keeps the whole history the development process and provides a the means for auditing the results and reverting to previous states if deemed necessary. Development can be observed and critized with the change tracking obtained from CVS. Another important facility is its means as a verifier and validator of development.

3.2.7 Build Practices

Okan group employs an automatically generated makefile to provide systematic builds. The makefile is included in the deployed package.

3.3 Configuration Status Accounting

CSA enables the team members to realize the status of the configuration items, and to properly communicate with other team members based on the information gathered by CSA. All team members will notified of the changes and they will not be detached from the project. CM reviews will be performed regularly o validate the current status of the project.

The team will use a change description form to convey the details about the modification. These forms will also be stored on an xml database so that all members will be able query the current status easily. These forms will be filled before commits to the CVS repository. The change description form consists of:

- name of the member committing the change
- version number of the checked out file
- description of change
- list of effected files and modules
- date of change (automatically generated)
- id of change description form (automatically generated)

An audit review file will also be placed on the CVS repository which includes the configuration items under auditing, the thoughts and review on these items.

Releases are made according to a schedule and a changelog of these releases which is bundled with the deployed package will carry information about the changes to the software.

3.4 Auditing

Auditing will be performed on the basis of requests of members during weekly meetings. It will be performed according to a list of criterias defined before auditing. These criterias will be based on the nature of auditing (whether it is functional, physical etc.). As for change requests and commits, describing documents are available for reference whether the desired functionality is accomplished. As stated previously these audits will also be saved. Audits are deemed mandatory before releases, snapshots and finalization of modules.

Without a proper documentation, the software is useless. In order to verify the quality of the documentation, regular auditing will be performed on the documentation.

4. Project Schedules and CM Milestones

Photokan project consists of independent modules whose finalization form a new milestone. Every two weeks, a module will be finalized. Software development and its time planning is a dynamic process so the living schedule may be changed to accomodate with the current situation. These milestones also indicate an integration and auditing process. After integration, an auditing will take place immediately to verify the success of integration. Even though milestones are predetermined for update of CMP, CMP can be updated any time if deemed necessary.

Table.2 Project Schedules and CM Milestones

09.03.2008	Delivery of CMP
11.03.2008	Deployment of project manager module
25.03.2008	Deployment of mosaicker and image registration module
25.03.2008	CMP Update
08.04.2008	Deployment of image processing module
22.04.2008	Deployment of DEM extractor and DEM displayer module
22.04.2008	CMP Update
06.05.2008	Deployment of orthophoto module

5. Project Resources

In order to maintain the CM activities, the team needs special software and experience. All of the members of Okan group are senior students and are experienced in development in C++, the language of implementation for Photokan. The necessary software consists of a CVS client since the team members will make extensive use of CVS. As mentioned early, tortoiseCVS, Eclipse IDE and the command line CVS client are the choice of the team. All of these resources are open for everyone and can be obtained easily. Every team member is acquintant with the rationale behind CVS, and acknowledges its value as a CM resource.

The website is another valuable resource that ensures communication with the audience of the software. Snapshots, demos and news about current status of the project can be obtained from the request.

6. Plan Optimization

Okan group has 4 members and members meet frequently so information is conveyed fast but a plan optimization is necessary between milestones to keep with the strict schedule and deadlines. After every milestone, plan optimization will be performed during the update of CMP.