MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

CENG 492

CONFIGURATION MANAGEMENT PLAN

MILSOFT-PHOTOGRAMMETRY LAB PROJECT

HANİFE ÜNAL

MERYEM AYAS

SERAP ATILGAN

SERRA SİNEM TEKİROGLU

March 9, 2008

Table of Contents

1 INTRODUCTION

This document is the Configuration Management Plan which defines the implementation of configuration management of PHOTOLAB project developed by Bad Sector team.

1.1 Purpose

Nowadays, almost all software engineers know that a problem cannot be identified until it is completely solved. The phenomenon behind this reality is the capability of the nature of the software development process to change. At the early stages of the development, the problem is usually innocent and straightforward. However, as the time passes, its capability of becoming a nightmare increases rapidly due to the missed schedules, customer's never-ending demands of new requirements, misunderstanding of the requirements, lack of communication between developers, bad software test and quality plans or many other reasons. The above reasons, leading the software development process to become a monster, show that it is impossible to foresee a problem and possible outcomes or side effects of the changes made to solve that problem. If the changes are not evaluated and made professionally, the result may even be a crash in the software. Therefore, developers should analyze the problem and possible changes in terms of predefined rules and policies.

In the light of the above information, these predefined rules and policies can be incorporated under Configuration Management Plan. The main purpose of this CMP document is providing and maintaining integrity of the software in every stage of its development. The CMP will identify the steps that the developers should follow in case of a change. In other words, CMP will ensure that who will do the change and how she will do it, how the change will be integrated to the whole project. CMP will also ensure that the change guarantees not to affect the other parts of the project badly, and it will be documented according to the standards stated in CMP. In this way, CMP will provide the elimination of risks which originate from developers, the sponsor company and the nature of the software development process.

1.2 Scope

The scope of this document is the identification of a top-level configuration management plan for Bad Sector team members during the development of the PHOTOLAB project. CMP will provide the control of releases and changes (with the help of Configuration Management Tools), and it will trigger the team to record and report the status of the project, auditing and reviewing. Apart from these prominent benefits, CMP will specify the organization of the CM framework and the responsibilities of the team members, the CM Tools used, the CM milestones and resources used in CM processes.

1.3 Definitions, Acronyms and Abbreviations

CM: Configuration management
SCM: Software configuration management
SCMP: Software configuration management plan

CMT: Configuration management tools
FTR: Formal technical review
CI: Configuration item
SCR: Software change request
SDT: Software development team
STT: Software testing team
CCT: Change control team
VCT: Version control team
RCT: Release control team
CMCT: Configuration management control team
CSA: Configuration Status Accounting
FCA: Functional Configuration Auditing
PCA: Physical Configuration Auditing

1.4 Document Overview

This document is divided into seven major sections namely the Introduction, The Organization of CM Framework, The Configuration Management Process, Project Schedules - CM Milestones, Project Resources, Plan Optimization and Appendix. The organization of this document is as follows:

- **Section 1– Introduction:** This section presents preliminary information concerning this document. The introduction gives information about the purpose and scope of this document, the organization of this document, and any references utilized in the assembly of this document.
- **Section 2– The Organization of CM Framework:** This section describes the organization, responsibilities of the teams on configuration management, tools which are used in configuration management process and infrastructure of the configuration management process.
- **Section 3– The Configuration Management Process:** This section describes the configuration management process to be used.
- **Section 4– Project Schedules - CM Milestones:** This section describes the Project Schedule and Configuration Management Milestones which provides the team members for working in a coordinated- communicated way and triggers them to be punctual in order to finish the project on time as a guide in project planning for CM activities.
- **Section 5– Project Resources:** This section simply identifies the resources used in configuration management process**.**
- **Section 6– Plan Optimization:** This section describes the optimizations made on planning of the project which is specified in Requirement Analysis Report.
- **Section 7- Appendix:** This section consists of the living schedule of PHOTOLAB project.

1.5 Document References

1 “configuration.management.ppt”
METU Computer Engineering, Ceng492, 2004
http://www.ceng.metu.edu.tr/courses/ceng490/Documents/

2 “CMPlanTemplate.doc”
Systems Engineering Process Office, Code 20203
Space and Naval Warfare Systems Center San Diego, FEBRUARY 21, 2006

http://sepo.spawar.navy.mil/CM_**Plan_Template**.doc

3 “CVS Concurrent Versions System”
http://www.nongnu.org/cvs/

4 “CONFIGURATION AUDITING”
TRAMM, Volume 3 Section 1
http://www.defence.gov.au/army/PUBS/Tramm/Volume%203/Section%201/V3-S1-C5.pdf

2 ORGANIZATION OF CM FRAMEWORK

2.1 Organization of SCM Teams

In order to make a good organization for the configuration management of the project, Bad Sector team members formed smaller SCM teams as stated below:

- Software Development Team (SDT)

  The team is responsible for the development of the whole PHOTOLAB modules specified in Detailed Design Report and they implement the change requests which come from CCT and STT. They will create releases and update the CVS whenever there is a significant change which triggers the developer to create a new version of the project. They will also create FTRs for the CCT whenever there is a doubt about the change before or after the change is implemented.

  The members of SDT are Hanife Ünal, Meryem Ayas, Serap Atılgan, and Serra Sinem Tekiroğlu.

- Chance Control Team(CCT)

  The team is responsible for the managing and monitoring of the CM activities, and they evaluate the SCRTs which come from SDT, SST and all of the control teams and then request, assign or reject the changes. The members of CCT are Hanife Ünal and Serra Sinem Tekiroğlu.

- Software Testing Team(STT)

  The team is the main source of the SCRs created during the implementation of PHOTOLAB. They will prepare test cases and will apply them until they find a bug or finish all of the cases successfully. The team will follow the testing procedure stated in detailed design report and will do the component, integration, interface, and performance testing as well as specifying pass/fail criteria. Having finished each testing, the team will prepare SCRs for the CTT (if needed) and publish the test results. The members of STT are Serap Atılgan and Meryem Ayas.

- Version Control Team (VCT)

  The team is responsible for the monitoring of the versions of PHOTOLAB. They will control the versions of the product from begging of the implementation to the end. The team will also do backups for the substitution of an older version of the product if there is a problem with the new

version. And they will report the current situation regularly. The member of the team is Serra Sinem Tekiroğlu.

- Release Control Team (RCT)

  The team is responsible for the monitoring and controlling of the current release of PHOTOLAB. The member of the team is Hanife Ünal.

- Configuration Management Control Team (CMCT)

  The team is responsible for the monitoring of CM activities and ensuring that the activities are all done according to the specified schedule. The team will also update the Configuration Management Plan and Living Schedule in case of a need. The member of the team is Meryem AYAS.

2.2 Responsibilities of the SCM Teams

Main responsibilities are listed below with the team responsible for that responsibility. CMCT coordinates the responsibilities and assigns teams for each.

- **Configuration Identification:**
  - Establish and follow release procedures to obtain Product Baselines for new version releases→ RCT
  - identify, document, archive, and track changes to system releases→ CCT
  - Coordinate assignment of identifying numbers for CIs and documents.→ CMCT
  - Coordinate release of software and associated documentation→ RCT
- **Configuration Control:**
  - Establish and use configuration controls for software, and documentation.→ CMCT
  - Establish and document configuration change control procedures.→ CCT
  - Place contents of Baselines and Developmental Configurations under configuration control→ SDT
  - When appropriate, generate executable load modules from controlled source code. → SDT
  - Prepare and maintain master of the currently active version of each CI until superseded by a new version. → RCT
  - Perform tests→ STT
  - Review, and validate Configuration Control→ VCT, RCT, CCT, CMCT
- **Status Accounting:**
  - Provide CSA implementing and recording → STT
  - Prepare status reports on change requests→ STT,SDT
  - Maintain an accounting of configuration item changes by tracking change requests→ CMCT
  - Review and Validate CSA→CCT

- **Auditing and Reviewing:**
  - Perform reviews of CM activities and products→ CCT
  - Review and update CM documentation as required for ensuring that current applicability is maintained. →CMCT
  - Report the need for auditing and reviewing→ SDT

2.3 Tools & Infrastructure

- **wxDev++:**

WxDev-C++ is an extension of Dev-C++ which is an open source IDE for C and C++. This program helps to create dialogs and frames for wxWidgets visually using a form designer.

- **CVS:**

CVS is a version control system and an important component of Source Configuration Management (SCM). Using it, one can record the history of sources files, and documents and their current versions as well as allowing the developer to see precisely the changes or what caused a bug between versions. It is useful in cases where several developers or teams want to each maintain their own version of the files because CVS can merge the changes and versions by giving the latest version to the developers. Therefore Bad Sector team's primary version control system is CVS.
There are two main CVS clients which Bad Sector team will use. First client is CVS client of Eclipse. This client is the main client that will be used in all CM activities since it is easy to use and practical. Therefore every member of Bad Sector team has Eclipse installed on her own computer and will use it regularly. The second client is the CVS client of METU Computer Engineering Department. The client will be used in case there is a problem with the CVS client of Eclipse, since it is difficult to use.

- **Infrastructure:**

The infrastructure of the CVS directory of Bad Sector team will be as follows:

- There are 3 main directories namely "Versions" and "Modules" and "Documents".
- Versions directory has sub directories which have the previous and current version of the product.
- Modules directory has three sub directories which belong to three main modules of PHOTOLAB. Each sub directory will maintain the classes or related files belonging to that directory.
- Documents directory has files in .txt format which either can be a SCR or CSA.

3 THE CM PROCESS

3.1 Identification

PHOTOLAB implementation is basically formed of modules. Bad Sector team specified the configuration items according to the modules and the parts which form these modules. As explained in CVS infrastructure, all of the modules have different directories, each directory has a tree like structure and all of the files related to that module will have path “XModule/ YDir/ ZFile”. Configuration Items of PHOTOLAB project is explained as follows:

- **Source Code:**

The main part of the implementation is the source code of the PHOTOLAB. Therefore, Bad Sector team specified more CIs than the other parts in order to give the necessary priority to that part, and the team identified each of the classes of PHOTOLAB as a CI. The CIs of the source code is:

- MainWindow Class
- BasicToolbar Class
- ProjectManager Class
- ProjectManagerWindow Class
- ProjectDialog Class
- EnhancementToolbar Class
- Statusbar Class
- FileSystemHandler Class
- HistoryWindow Class
- ImageData Class
- PhotogrammetryManagerToolbar Class
- PhotogrammettryManager Class
- DEM Class
- Orthophoto Class
- Mosaic Class
- SuperResolution Class

These are all of the classes of PHOTOLAB which are decided to be CIs for the CM process. Each class has a source file (.cpp) and a header file (.h). However, in case of a need, classes may have more than a source file or header file.

- **Data**

In order to have a good visualization of the software (buttons, toolbars, windows, or boxes), Bad Sector team will use some icons and these icons form the Data CIs of PHOTOLAB. Each icon will have path "XModule/IconsDir/XIcon".

- **Documentation**

Just like every other good software projects, PHOTOLAB has a good documentation. Bad Sector team has published a lot of documents in previous semester, and it goes on to publish new ones. The documents which have been published so far are listed below:

- Project Proposal
- Requirement Analysis Report
- Initial Design Report
- Final Design Report
- Living Schedule
- Weekly Reports.

Here comes the Configuration Management Plan. After the CMP, the team will prepare a Test Specifications Report and a User/Developer Manual. All of these documents can be reached from the Bad Sector team's website which is updated regularly.

- **Baselines**

Baselines are basically the tasks which have to be done at a specific time in the project, meaning milestones of the project. Bad Sector team had prepared a Gantt chart identifying the schedule and milestones of PHOTOLAB in the previous semester. All of the team members worked synchronously with the schedule, and the milestones which belong to the semester have been achieved. This semester the team has prepared a living schedule in addition to Gantt chart and specified the baselines, milestones, as CIs which will be tracked, audited, retained and version controlled in CVS. The baselines are:

- Configuration Management Plan

- Implementation of PHOTOLAB modules and their integration to the whole system
- First Snapshot Demonstration
- Test Specification Report and Testing
- Preparation of User & Developer Documentation
- Final package and Final Demonstration/Competition.

➢ **CI Versioning Conventions:**

Versioning is the process of assigning either unique version names or unique version numbers to unique states of the software and has an important value in configuration management. Therefore our team will also use software versioning which assign numbers to versions of the product in increasing order which means the bigger the numbers, the later the version.

Bad Sector team will use the most common software versioning scheme

"major. minor [.build[.revision]]". And PHOTOLAB's first released version will be 1.0. Numbers smaller than 1.0, will indicate the alpha or beta versions.

3.2 Configuration Management and Control

3.2.1 SCRs

During the implementation of PHOTOLAB any member of the team may come up with a change request if she is not satisfied with the finished part of a work or there is a bug in somewhere in the current version of the product. The main team which will request a change is STT. The other teams SDT, VCT, RCT also may come up with a change request. The change request has to be in a predefined format so that every member of the team can understand the reason of the change, the parts which will be affected by the change, and the deadline of the change. Bad Sector team specifies a SCR format as follows:

"SCR[SCRID].txt"

SCR ID: SCR1.0
SCR DATE: 08.03.2008
SCR DEADLINE:7 days
SCR PRIORITY:9 (from 1 to 10)
NAME OF THE MEMBER REQUESTING: Meryem AYAS
NAME OF THE RESPONSIBLE MEMBER FOR THE IMPLEMENTATION: Serra Sinem TEKIROGLU
NAME OF THE RESPONSIBLE MEMBER FOR THE TESTING: Serap ATILGAN
RELATED SOURCE NAME: PHOTOGRAMMETRY MODULE/MOSAIC Class/ Mosaic.cpp
RELATED VERSION OF THE PRODUCT: PhotolabV1.0
EVALUATED:YES
IMPLEMENTED:NO
TESTED:NO

DETAILED CHANGE DESCRIPTON:
Ransac function of the mosaic class doesn't give a satisfactory number of homography matrices to evaluate and find the best homograpy matrix. As a result the best homography matrix can't give the resultant image correctly. It should produce more than fifty homography matrices.

The SCRs will be put under the directory CVS/Documents and the member who requests a change will inform the CCT with an email or telephone call in case of an emergency. Normally the team responsible for the SCRs will check the CVS to learn whether there is a new SCR or not and the team will evaluate the SCRs and assign one or more member for the change if the change is not rejected. After the change is made correctly, SCR id and the author for the change will be documented at the beginning of the source file.

3.2.2 SCR Evaluation

SCR evaluation will be at the meetings held in once a week. The team for the evaluation is of course the CCT and SDT. CCT will inspect the SCRs and will put the SCR to the vote. If the SCR is accepted, then CCT will put a deadline according to its priority and assign one or more members for the implementation, integration and testing of the change.

3.2.3 SCR Implementation and Testing

After the evaluation, SCR will be passed to SDT team members who are assigned with the implementation. They will implement the change and integrate it up to the deadline specified by CCT in SCR. After the integration they will pass the SCR to STT and they will do the testing on the new version of the source. After all of the test cases have been passed, the STT will fill the "TESTED" section of SCR as "YES" and will inform the VCT to update the CVS in order to get the final version.

3.3 Status Accounting

Status accounting becomes an important concept as the project advances. PHOTOLAB is a big project, consisting large source files, which must be tracked in every stage of the development. Therefore, Bad Sector team decided to keep a log informing the members about the status of the current version of the product. Hence, every member of the team will be aware of the changes made and status of the software. The file decided to be the log file is named as "CSADate.txt ", and will be under the directory

CVS/Documents in CVS. All of the CIs specified by the team will be tracked and the changes made in these CIs will be logged to that file. The format of the file is as follows:

CSA[dd.mm.yy].txt

Previous Version:
Current Version:
/* Below are the change request logs pointing the SCRs which can be tracked by ids in CVS/Documents*/
SCR ID:
SCR ID:
..
..
..
/* Below are the CIs which will have been finished by the date of the log file.*/
Finished Modules:
Finished Classes:
Last Achieved Baseline:
/* Below are the awaiting implementation details.*/
Awaiting Baseline:
Awaiting Features/Modules/Classes to be Implemented by the next week:

3.4 Auditing

Auditing is an important part of the CMP since it provides the developers the way to increase the quality of the product. Therefore Bad Sector team specified an auditing program which will be conducted both in specific days and in weekly meetings. Bad Sector team meets twice a week. In one of the meetings we will just talk about the SCRs belonging to that week and evaluate the SCRs as well as continuing the implementation all together. In the second meeting, we will be doing audits. The main source which will be a guide in these audits is the CSA[Date].txt. The team will examine the log file and evaluate the current situation of the product, the changes, what has been done so far and what will be done in the following week.

The second type of the audits is the audit which will be conducted in one or two days before a specific day. These specific days are the days just before milestones, identified by the Gantt chart of the team and the course syllabus namely, First Snapshot Demonstration, First Release, Second Release and the Final Package Demonstration. To specify in detail, this second type of the audits will be divided into two categories which are Functional Configuration Audits (FCA) and Physical Configuration Audits (PCA). In the days except the day before Final Package Demonstration, we will be doing FCA. Usually, the FCA represents a review of the item's performance, to ensure it not only meets the specification but that there are no unintended functional characteristics. These FCAs, Bad Sector will conduct, include:

- reliability, availability and maintainability tests and trials
- user trails
- interface checks and tests

- software testing, including member by member verification and validation.

The day before the Final Package Demonstration we will be doing both FCA and a PCA. Since a PCA involves the comparison of the new product with the design documentation to ensure the physical characteristics and interfaces conform to the product specification and it shows that the development is complete and the product has been built in accordance with its functional and non functional requirements, we will be doing a PCA before representing our final package. The PCA will include:

- review of FCAs to verify incorporation of CIs
- review of specifications
- examination and recording of deviations
- examination of documentation.

4 PROJECT SCHEDULES AND CM MILESTONES

Time planning is one of most important building stones of CMP since it reduces the risk of missing schedules due to the developer, sponsor company or development cycle related reasons. In the light of that reality, Bad Sector team has prepared Gantt charts and a living schedule of the project. These documents give a detailed time specification of the development of PHOTOLAB project. Hence, the team members will never fall behind the schedule and achieve the milestones on time. The milestones specified by Bad Sector team are Baselines, Module Completions, Finished Part Implementations and Documentation. Below are the milestones and their deadlines:

File System Module Implementation	06.03.08
Configuration Management Plan Report	09.03.08
First Snapshot Demonstration	13.03.08
File System Module Integration& Testing& Optimization	25.03.08
GUI Module Implementation	11.04.08
Software Test Specification Report	17.04.08
GUI Module Integration& Testing & Optimization	25.04.08
Photogrammetry Module Implementation	20.05.08
Photogrammetry Module Integration& Testing &O.	25.05.08
Whole System Integration & Testing & Optimization	10.06.08
Final Package Demonstration	13.06.08

Note: There will be two product releases. However, their deadlines are not informed yet. They will be included in the above chart in the updated version of CMP.

5 RESOURCES

Resources which will be used by Bad Sector team are stated below:

- All of the Bad Sector team members
- Adem Mulayim from sponsor company MILSOFT

- Supervisor Murat Yükselen
- CVS
- WxDevC++
- OpenCV Library & its documentation on the web
- WxWidgets Toolkit & its documentation on the web
- CMP
- All of the documentation which have been prepared so far.

6 PLAN OPTIMIZATION

Configuration Management Plan will help Bad Sector team in the development cycle of PHOTOLAB. Since the development cycle is an incremental period, this document has to be updated according to the changes and milestones. This is the first version of CMP of PHOTOLAB project and in order to make CMP a good resource, the document will be updated in every achieved deadline stated in the above chart by responsible team, CMCT. In case of an urgent need of change, the team will not wait for the deadline comes and will update the document according to the change.

7 APPENDIX

TASK DESCRIPTION	DURATION	START	FINISH	ASSIGNED TO
GUI MODULE	60 days	25.02.08	25.04.08	All
BasicToolbar Class Implementation	15 days	04.03.08	19.03.08	Serap
EnhancementToolbar Class Implementation	20 days	12.03.08	22.03.08	Meryem
ProjectManager Class Implementation	15 days	03.03.08	18.03.08	Hanife
MainWindow Class Implementation	45 days	25.02.08	11.04.08	All
Testing & Optimization of Module	10 days	15.04.08	25.04.08	Serra, Meryem
FILE SYSTEM MODULE				
FileSystemHandler Class Implementation	10 days	10.03.08	20.03.08	Serra
ImageData Class Implementation	5 days	01.03.08	06.03.08	Hanife
Testing & Optimization of Module	5 days	20.03.08	25.03.08	Serap, Hanife
PHOTOGRAMMETRY MODULE				
PhotogrammetryManagerToolbar Class Implementation	5 days	25.03.08	30.03.08	Serra
PhotogrammetryManager Class Implementation	80 days	25.02.08	15.05.08	All
Testing & Optimization of Module	5 days	15.05.08	20.05.08	Meryem
MOSAIC SUBMODULE				
Method Searching	3 days	03.12.07	06.12.07	All
Mosaic Class Implementation	85 days	06.01.08	01.04.08	All
Testing and Optimization	10 days	02.04.08	12.04.08	Serra
DEM SUBMODULE				
Method Searching	5 days	20.03.08	25.03.08	Serra, Meryem
DEM Class Implementation	35 days	26.03.08	30.04.08	All
Testing and Optimization	5 days	30.04.08	05.05.08	Hanife, Serap
ORTHOPHOTO SUBMODULE				
Method Searching	5 days	15.04.08	20.04.08	Hanife, Serap
Orthophoto Class Implementation	30 days	20.04.08	20.05.08	All
Testing and Optimization	5 days	20.05.08	25.05.08	Serra, Meryem
WHOLE SYSTEM TEST & OPTIMIZATION	15 days	25.05.08	10.06.08	All
OTHER				
Configuration Management Plan	5 days	04.03.08	09.03.08	Meryem
Web site	105 days	25.02.08	10.06.08	Hanife