MIDDLE EAST TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

CENG492 CONFIGURATION MANAGEMENT REPORT

KORSAN YAZILIM
# Table of Contents

1. Introduction ................................................................. 3
   1.1 Purpose of CMP ......................................................... 3
   1.2 Scope of Document .................................................... 3
   1.3 Definitions, Acronyms, Abbreviations ................................. 3
   1.4 Document References ................................................ 4
   1.5 Document Overview .................................................. 4
2. The Organizations CM Framework ........................................ 4
   2.1 Organization ........................................................... 4
   2.2 Responsibilities ...................................................... 5
   2.3 Tools and Infrastructure ............................................. 5
3. Configuration Management Process ...................................... 5
   3.1 Configuration Identification ....................................... 5
   3.2 Configuration Management and Control ........................... 7
   3.3 Configuration Status Accounting .................................. 8
   3.4 Auditing ............................................................... 8
4. Project Schedules and CM Milestones ................................. 9
5. Project Resources .......................................................... 9
6. Plan Optimization .......................................................... 9
1. Introduction

This document is the configuration management plan of Context Aware User Interface project, performed by the group “Korsan Yazılım” sponsored by Aselsan.

1.1. Purpose of CMP

Yaver is an application which includes many functionalities and it will become very complex while we work on it during semester. As we work on the project, in order to protect our implementations from misuses, to prevent critical mistakes and errors which can happen during the process, to communicate with other members about what they did about project and importantly to have standardization about project, we prepared this CMP. Our intended audiences are us and our assistants and lecturers of CENG492.

1.2. Scope of the Document

This document is for specify CMP of Korsan Yazılım group. This report will be presenting organization, planning and roles of our team members in the process of implementing this project. There will be some specific plans for specific events.

1.3. Definitions, Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMP</td>
<td>Configuration Management Report</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>SVN</td>
<td>Sub-Version</td>
</tr>
<tr>
<td>SCR</td>
<td>Software Change Request</td>
</tr>
<tr>
<td>CCT</td>
<td>Configuration Control Team</td>
</tr>
<tr>
<td>SDT</td>
<td>Software Development Team</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>JDBC</td>
<td>Java Database Connectivity</td>
</tr>
<tr>
<td>JMF</td>
<td>Java Media Framework</td>
</tr>
<tr>
<td>DDR</td>
<td>Detailed Design Report</td>
</tr>
<tr>
<td>MySQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>Javadoc</td>
<td>Documentation Generator</td>
</tr>
</tbody>
</table>

1.4. Document References

[1] Software Configuration Management lecture slides from Ceng492 Spring 2011
Website


1.5. Document Overview

Introduction part of document includes purpose and scope of the configuration management, some definitions, acronyms and abbreviations and document references. In second part, organization and responsibilities of the project team on CM are given with the tools and infrastructure of the process. Then configuration management process, plans, significant dates, resources and methods to be used will be explained.

2. The Organizations CM Framework

2.1. Organization

CM activities are handled by the following groups:

- **Configuration Management Team:** The duty of this team is taking care of the CM organization by updating it every time when necessary.

- **Configuration Control Team:** This team will supervise all the activities of other groups. Nevertheless, reviewing SCRs, accepting or rejecting SCRs and monitoring SCRs are the main responsibilities of CCT.

- **Software Development Team:** The main task of software development team is to implement all modules of Yaver. In addition; creating releases, updating source code via SVN and fixing bugs are the responsibilities this team.

- **Release Control Team:** This team is going to monitor versions of Yaver. Furthermore, it is responsible for merging different branches of Yaver.

- **Testing Team:** This team is responsible of testing and debugging of the implementations produced by SDT. Checking if the requirements correspond to the
implementations, stating the problems and of course returning feedback (SCR) to software development team are in the duties of testing team.

2.2. Responsibilities

Each member of Korsan Yazılım is responsible from all the organizational teams described above. Any single change in the CMP is to be first argued in CCT and to be committed afterwards.

2.3. Tools and Infrastructure

During the development and management of Yaver, the following tools will be used.

**Net Beans:** Refers to both a platform framework for Java desktop applications, and an integrated development environment (IDE) for developing with Java, JavaScript, PHP, Python, Ruby, Groovy, C, C++, Scala and Clojure [3]. We use NetBeans for developing all modules of the projects namely the camera operations, image processing, server since NetBeans offers a good environment for developing with these technologies such as JMF, JDBC.

**SVN (Subversion):** In software development, Subversion (SVN) is a version-control system [4]. Developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation.

**Trac:** Trac is an open source, web-based project management and bug-tracking tool [5]. Trac allows hyper linking information between a computer bug database, revision control and wiki content. It also serves as a web interface to a version control system, in our case SVN. With the help of the Trac, we will assign tasks to the members of SDT and also SCRs are raised via Trac.

**Web Page:** All documents and project progress can be seen via web page.

3. Configuration Management Process

3.1. Configuration Identification

At this section we will introduce some information about source code, database actions and documentation.
3.1.1. Source Code

The source code files are kept in an area hosted by DROPBOX and the repository of the SVN hosted by department synchronously. We will use 2 repositories to protect source files against possible hardware or network failures. All files can be accessible by all members of the team to add new files or update existing files. Moreover, assistant and instructors can also have permission to see current changes. All of the implementation is done in Java Programming Language with adding extra libraries and tools.

In order to provide maintainability and understandability of code to developers, we have defined some simple rules.

3.1.1.1. Function Names:

Each function should start with lowercase letters. If a function name consists of more than one word, each word concatenated and starting letter of each word will be capital.

Example:
function() , myFunction(), findAllPossibleMapsFromServer() etc… are valid.
Get(), getcoordinate(), set_label(), ISVALID() etc… are not valid.

3.1.1.2. Variable Names:

Since poorly-named variables make code harder to read and understand, it is better to determine a general name convention for this project. As a result, all variables begin with a lowercase letter, and if consisting of multiple words, each are connected with underscore ( _ ) character. Since same named local variables hide global variables, starting with underscored name is better. Loop iterators ( i,j,k … ), and intermediate variables declared as anything developer wants.

Example:
coordinate, current_coordinate, button_size, _contrast(only valid if global) etc… are valid.
x (it is not math lesson), ObjectId, reachDate, Priority etc… are not valid.

3.1.1.3. Class Names:

All classes begin with capital letters. If a class name consists of multiple words, each starting letters of words should be capital. Abstract classes begin with underscore.

Example:
ToolBar, MapExplorer etc… are valid.
map, messageSearcher, Assignment_Lister etc… are not valid.

3.1.1.4. Comments:
Increasing the understandability of the codes, we write comments after some lines. We will use Javadoc style commenting. Moreover, developers can add extra commands and tricks, if he doesn’t be satisfied with Javadoc.

3.1.2. Database Management

Both sent and received messages, and assigned assignments are stored in database. We will reach them or add new items by using MySQL. The reason is that all of the team members are experienced about MySQL queries in database interactions.

3.1.3. Documentation

Design reports of this project

- Requirement Analysis Report (√)
- Initial Design Report (√)
- Detailed Design Report (√)
- Revised Design Report (√)
- Configuration Management Plan Report (√)
- Test Specification Report (×)

Development Process Reports

- Weekly Reports (☑)

User Manuals

- Installation Manual (×)
- Development Guide (×)

(√)’s completed, (×)’s not completed, (☑)’s still continue.

3.2. Configuration Management and Control

The process for notifying, evaluating and solving possible configuration change requests can be analyzed in four stages.
3.2.1. Configuration change request

Minor requests are going to be directly imposed in to the project and handled by SVN. If there is a major change in the project, the process is going to be managed by Trac system. Each request is going to be represented as a ticket. This ticket assigned to one of the developer as soon as possible and every member of the group is going to be informed about new ticket situation.

3.2.2. Configuration change evaluation and approval

Before evaluation part, ticket-assignee developer has to accept that if change is related to his own part. If change is not related to his area, he has to reject and reassign ticket to another developer. After accepting ticket, if developer has confusing about change, we are going to talk possible solution in weekly and extra meeting time. After applying changes, assigner is going to control and test it and alter ticket status to be solved.

3.3. Configuration Status Accounting

Keeping track of status of the project becomes more important as project improves and flourishes. In order to provide coordination and synchronization between modules of project and works of group members, configuration status accounting is crucial. By using SVN, we literally solve this issue, because of that any changes on source code or project will be saved with detailed background information such as who, when did this change to project. So, we guarantee that we will not lose any crucial information during the process. We will use comments and explanations to SVN commits to make changes more understandable. We all will be alert for any data loss or conflict. Therefore, we ensure that our progress and improvement along the project will be safe and healthy.

3.4. Auditing

Communication between group members of Korsan Yazılım is done via weekly meetings, instant messengers and phones. In each weekly meeting, each group member states what he has done in the previous week. Problems that may possibly be faced are discussed and new solutions are suggested by group members in order to follow the schedule strictly. Moreover, new tasks are assigned to each group member for the following week. Korsan Yazılım not only discusses problems together but also does implementation together, but in
some cases duties are distributed among group members. Critical parts of the implementation are done together by arranging extra meetings during the week.

Korsan Yazılım has its own tradition for reducing implementation bugs. Each group member checks his work before submitting any lines of code to SVN. This removes almost all of the bugs which may occur at least for the trivial cases. Each group member also checks changes done by other group members and gives immediate feedbacks if any mistake is seen.

4. **Project Schedules and CM Milestones**

Living schedule and milestones to be completed are in the webpage of Korsan Yazılım. Having implemented the prototype of Yaver at the end of the first semester, increasing user-friendliness of the user interface and adding context-aware changes according to the three states of motion are the main features which will be in the final release of Yaver.

The milestones that will be followed are as follows:

- First Build – 20 April
- Second Build – 13 May (tentative)
- Final Release – 27 May
- Documentation and manuals – 10 June

5. **Project Resources**

The resources of the projects can be listed as follows:

- Korsan Yazılım Webpage
- Documents prepared(SRS,DDR)
- SVN
- NetBeans
- Eclipse
- Trac
- Dropbox
- Java Libraries

6. **Plan Optimization**

Configuration management plan will guide us during the implementation of Yaver. Each group member's tasks are stated in the living schedule. According to living schedule,
each group member is responsible for doing his tasks. To reduce the risk of missing milestones, living schedule is arranged to tolerate small delays in any case of problems. If any delay, or changes occur in the progress of Yaver, these will be reflected to the living schedule and will be seen in the web page of Korsan Yazılım.