

KAROSHI

Configuration
Management
Plan

1559897 İlker ARGİN

1560499 Meryem SAĞCAN

1630540 Fatma AKINCI

1560036 İsmail Can COŞKUNER

Contents

- 1. Introduction..... 2
 - 1.1. Purpose of Configuration Management Plan..... 2
 - 1.2. Scope of the Document..... 2
 - 1.3. Definitions, Acronyms and Abbreviations 2
 - 1.4. Document References 3
 - 1.5. Document Overview 3
- 2. The Organizations CM Framework 3
 - 2.1. Organization..... 3
 - 2.2. Responsibilities 4
 - 2.3. Tools and Infrastructure 4
- 3. Configuration Management Process 5
 - 3.1. Identification 5
 - 3.1.1. Source Code 5
 - 3.1.2. Data 5
 - 3.1.3. Documentation 5
 - 3.2. Configuration Management and Control 6
 - 3.2.1. Software Change Requests 6
 - 3.2.2. Software Change Evaluation 6
 - 3.2.3. Software Change Implementation 6
 - 3.3. Configuration Status Accounting..... 6
 - 3.4. Auditing 7
- 4. Project Schedules and CM Milestones 7
- 5. Project Resources..... 7
- 6. Plan Optimization 7

1. Introduction

1.1. Purpose of Configuration Management Plan

MAP-MET is a project in which many different modules work together and communicate with each other. In other words, it is a complex system in which a lot of components are integrated to do some service. Any modification made by a developer on a single entity should be notified to all other developers. That is why, CMP report is essential to prevent confusion among developers of the product. CMP allows our team to integrate changes to the project easily, since it is supposed to reduce the side effects of the changes. Furthermore, it is for sure that if a member of the project makes a change, CMP will help the members to apply these changes without errors.

1.2. Scope of the Document

The scope of this document is about the identification of configuration management plan for project MAPMET of Karoshi SDT. The document includes the organization of the team, Configuration Management Process being used and the schedule of the project. Versioning, differencing work product and reporting on all are also included in this document. The related audience for this document is Karoshi SDT, our assistant, instructors and sponsor company ASELSAN.

1.3. Definitions, Acronyms and Abbreviations

MAP-MET: MAP-Military Enhanced Technology

SDT: Software Development Team

SCR: Software Change Request

SVN: Subversion

CMP: Configuration Management Plan

CI: Configuration Item

CSA: Configuration Status Accounting

MTJ: Mobile Tools for Java

1.4. Document References

- [1] IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998
- [2] Configuration Management Plan Presentation, METU Computer Engineering, Spring 2011
- [3] <http://www.eclipse.org/>
- [4] <http://subversion.tigris.org/>
- [5] <http://trac.edgewall.org/>
- [6] <http://senior.ceng.metu.edu.tr/2011/karoshi>

1.5. Document Overview

This document is divided into six sections, the Introduction, the Organization, the Configuration Management Process, Project Schedule, Project Resources and Plan Optimization. In the introduction part we explained what is the meaning and purpose of a CMP and give the abbreviations and the references used. The organization of the team, the responsibilities of team members and tools used during the project development can be seen in the second section. The CM Process section gives information about the identification process, tools and practices for Management and Control, Configuration Status Accounting and Auditing. Project Schedules and CM Milestones are given in the 4th section. The resources needed during CMP are explained in 5th section. Finally we conclude with the methods to be followed for optimizing CMP.

2. The Organizations CM Framework

2.1. Organization

All the members of Karoshi contribute to CM to develop a successful project and have equal rights during the project management process. The teams that are required to manage the process are:

Software Development Team:

This team is mainly responsible for implementing the modules and making the changes that are requested by Testing Team. Releasing a version is also a duty of this team.

Testing Team:

This team is responsible of testing and debugging of the implementations produced by SDT. Second purpose of this team is to give feedback about the modules and to give SCRs if necessary.

Configuration Control Team:

This group has the duty of supervising the other three teams. To review the SCRS, accept or reject, and monitor them is the main responsibility of this team.

Configuration Management Team:

Configuration Management Team is responsible for maintenance of the CM organization. Hence, this group will keep the CMP up to date.

2.2. Responsibilities

Since each member of Karoshi is also a member of CCB the same responsibilities are taken by all the members and these responsibilities are conforming to CM schedule, commenting about changes before committing resources through SVN, and e-mailing people about SCRs.

2.3. Tools and Infrastructure

Eclipse IDE:

Eclipse employs plug-ins in order to provide all of its functionality on top of (and including) the runtime system, in contrast to some other applications where functionality is typically hard coded. MTJ plug-in for Eclipse will be used and all the codes for mobile device side will be written with the help of this IDE.

TRAC and SVN:

Trac^[4] is an enhanced wiki and issue tracking system for software development projects. It provides a Graphical front end to SVN where diffs in files can be obtained. It cannot update the SVN repository. It simply provides a Project Management interface, wiki, ticketing system, and SVN front end.

SVN^[5] is a free/open-source Version Control System. Subversion (SVN) manages files and directories over time. Files are placed into a central repository and every change every made to the files or directories are remembered.

3. Configuration Management Process

3.1. Identification

The CI's are so important to identify the project and these items are categorized into 3 categories: source Code, data, and documents.

3.1.1. Source Code

The source code files are kept in the repository of the SVN hosted by Department of Computer Engineering, METU resulting in access to all members of the team adding, updating source files and committing afterwards.

Each developer has to write proper comments in source codes. Variable names in the source codes have to be meaningful. Otherwise, it will be hard to keep track of the code. Moreover, some standards concerning the programming language used will be obeyed. (Class names in JAVA, variable names in java, the use of upper cases, etc.)

3.1.2. Data

XML files keeping some messages or location information will be named according to their functionalities. Map data will be kept in a single image in the mobile device.

3.1.3. Documentation

In our project, documentation is a mandatory CI. The documents created so far are the following:

Reports:

- Project Proposal
- Requirements Analysis Report
- Initial Design Report
- Detailed Design Report
- Configuration Management Plan

Web Documents:

- Web page

Development Process:

- Weekly Reports

After a while, Test Management report will be prepared. After the product is ready, User Manual will be written for the end-product.

3.2. Configuration Management and Control

3.2.1. Software Change Requests

Minor SCRs are directly added to the system and handled by SVN and requires no extra information. Major SCRs are controlled by the TRAC system. In this system a SCR consists of team member name, description, date, deadline, related module, priority, and version. When a SCR reported a ticket is opened by TRAC and it can be seen by all the team members.

3.2.2. Software Change Evaluation

Since Buğra Özkan (Teaching Assistant of Karoshi) has also access to the TRAC ticketing system we prefer discussing SCRs in TRAC system. After a new SCR is made by a member, at the next meeting with TA the SCR is discussed elaborately and the decision is taken. Then, either a ticket will be assigned to a member or will be refused

3.2.3. Software Change Implementation

If SCR is approved after an evaluation, all the possibly affected CI's will be detected and the related changes will be applied based on detected CI's. Then the version of the CI will be updated by committing the related resources through SVN.

3.3. Configuration Status Accounting

CSA just exists because accounting the status of the development at every little change is crucial in CMP. When a developer updates a source file in SVN, s/he gets to inform all the other members and the teaching assistant as well about what s/he added or changed. Also s/he gets to log the reason of the change clearly and s/he gets to be sure that the change is not conflicting with the other modules etc. This provides the development to be followed easily. Living Schedule is updated as well in regards of the accounting mechanism

3.4. Auditing

Communication between members is done by weekly meetings, SMS, TRAC system. These communications ensures the auditing. Each weekly meeting and its content are organized in the previous weekly meeting. All tasks are assigned to members and reminded via TRAC system and weekly meetings. The requests posed at the meetings are evaluated and applied if approved. During meetings, any disagreement is solved with finding an optimized solution. New researches are assigned to members in case of short term unforeseen problems. In addition to weekly meetings, extra meetings are arranged according to baselines.

4. Project Schedules and CM Milestones

The living schedule with all tasks and milestones to be completed will be in the website of Karoshi ^[6]. We have divided the development in components to ease workload sharing. These separated tasks can also be found at the living schedule. The milestones of MAP-MET are the following:

- First Development Snapshot, Demo: March 29, 2011
- First Release, Demo: May 10, 2011
- Final Release, Demo: May 23, 2011

5. Project Resources

- SVN : Subversion Control System
- Eclipse IDE : Development Environment
- TRAC: Issue Tracking System
- Web Site: Project Development News

6. Plan Optimization

We divided our CM responsibilities between our five members according to their interests. Each member is responsible for himself. We mostly communicate via phone call. In addition, we are making regular weekly meetings for more control over the group and also we have weekly meetings with our assistant. Moreover, we control our process with small demos for consistency of our project.