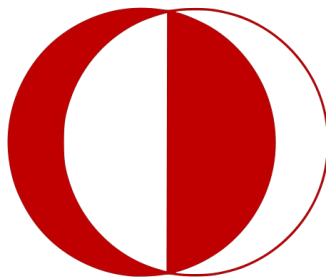


CONFIGURATION MANAGEMENT PLAN



FIZAN

MACROHARD

INDEX.....	2
1 INTRODUCTION.....	3
1.1 PURPOSE OF CONFIGURATION MANAGEMENT REPORT.....	3
1.2 SCOPE OF CONFIGURATION MANAGEMENT PLAN.....	3
1.3 ACRONYMS.....	4
2 CONFIGURATION MANAGEMENT FRAMEWORK ORGANIZATIONS.....	4
2.1 THE ORGANIZATION OF THE SCM TEAM.....	4
2.2 THE RESPONSIBILITIES OF THE SCM TEAMS.....	5
2.3 TOOLS AND INFRASTRUCTURE.....	6
3 CONFIGURATION MANAGEMENT STEPS.....	7
3.1 IDENTIFICATION.....	7
3.1.1 SOURCE CODE.....	7
3.1.2 WEB SIDE CODE.....	7
3.1.3 DOCUMENTS.....	7
3.2 CONFIGURATION MANAGEMENT AND CONTROL.....	9
3.3 CONFIGURATION STATUS ACCOUNTING.....	9
3.4 REVIEWS.....	9
4 PROJECT SCHEDULES AND CM MILESTONES	9
5 PROJECT RESOURCES.....	10
6 PLAN MAINTENANCE.....	10

1 INTRODUCTION

1.1 THE PURPOSE OF THE CONFIGURATION MANAGEMENT REPORT

There is a big potential in the Software market in our country. However software companies have not huge giro due to their unplanned works. Important part of these companies could not complete their projects successful and important part of handicaps while developing project is outcome of their documentless and unplanned disorganized works.

Changing project has some unwanted effects due to lack of coordination. Coordination in a big project seem difficult, however if configuration management report were used, coordination will not be a problem. Configuration management report supplies enough documentation and evaluation for the projects.

Our project Fizan will have some changes. In these times we will use this document to supply integrity. This document is also changeable with our project. In light of this document, we will see problems, we will fixed it and then afterwards we will change this document to see our new roadway.

1.2 THE SCOPE OF THE CONFIGURATION MANAGEMENT PLAN

This document's main role is software configuration management, means to supply Fizan's flexibility and maintainability. So, in every week we will try to change this document according to our changes and their needs. With little eye touches to this document, we know that we will see our roadway. In this document, roles of the Macrohard members can be seen.

There are some milestones for our project. We tried to make parallel working possible on our project Fizan. To control these milestones and manage it, we have some tools which make our coding easy. Our department supply us SVN and Trac tools. Addition to these, we

use Google group and web page to make communication perfect. These management styles are covered in this document.

1.3 ACRONYMS

VCT: Version control team

RCT: Release control team

FTR: Formal technical review

CI: Configuration item

CMUT: Configuration management updates team

TT: Testing team

CM: Configuration management

SDT: Software development team

SCM: Software configuration management

SCMP: Software configuration management plan

CT: Configuration teams

CCT: Change control team

CMT: Configuration management tools

CMR: Configuration management report

2 CONFIGURATION MANAGEMENT FRAMEWORK ORGANIZATIONS

2.1 THE ORGANIZATION OF THE SCM TEAM

Software Development:

- Kürşat Aksakallı
- Kemal Beşkardeşler
- Uğur Altun
- Ömer Faruk Çelik
- Zübeyir Aktürk

Testing:

- Kürşat Aksakallı
- Kemal Beşkardeşler

Change Control Team:

- Kürşat Aksakallı
- Kemal Beşkardeşler

Version Control:

- Uğur Altun

Release Control:

- Ömer Faruk Çelik

Configuration Management Update Team:

- Uğur Altun
- Ömer Faruk Çelik

2.2 THE RESPONSIBILITIES OF THE SCM TEAMS

Macrohard team has 5 members and each of them is part of a SCM team. Responsibilities of teams are:

Software Development Team: This team write codes, develop Fizan. Coding in 3 place, in Server with Java, in clients with Android SDK, in Web side it will be some dynamic language such as PHP or ASP.

Testing Team (TT): This team make tests to find bugs in Fizan. To make better tests, they prepare some cases and results which must occur.

Change Control Team (CCT): This team control changes and their effects, and inform other teams. CM activities will be accepted or rejected by this team.

Version Control Team (VCT): This team will control versions of Fizan. This team is responsible for merging different branches of Fizan.

Release Control Team (RCT): This team make plans about releases.

Configuration Management Update Team (CMUT): This team is responsible for updating configuration management plan and living schedule.

Also there are some responsibilities which everyone is responsible:

- Controlling trac, mail, google group and project's web page regularly.
- Following updated CM schedule and obeying deadlines.
- Informing other team members about his progress.
- Obeying team conventions about commenting and code writing.
- Committing written code regularly. These codes must be apparent and well commented.

2.3 TOOLS AND INFRASTRUCTURE

As tools, we will use SVN to make coding and versioning easy. SVN is a tool supply some advantages to the coder. There are some advantages of SVN, some of these are;

- Commits are true atomic operations. Interrupted commit operations do not cause repository inconsistency or corruption.
- Renamed/copied/moved/removed files retain full revision history.
- Directories, renames, and file metadata (but not timestamps) are versioned. Entire directory trees can be moved around and/or copied very quickly, and retain full revision history.
- Versioning of symbolic links.
- Native support for binary files, with space-efficient binary-diff storage.
- Apache HTTP Server as network server, WebDAV/DeltaV for protocol. There is also an independent server process that uses a custom protocol over TCP/IP.
- Branching and tagging are cheap operations, independent of file size, though Subversion itself does not distinguish between a tag, a branch, and a directory

- Natively client/server, layered library design.
- Client/server protocol sends diffs in both directions.
- Costs are proportional to change size, not data size.
- Parsable output, including XML log output.
- Open source licensed — "CollabNet/Tigris.org Apache-style license"
- Internationalized program messages.
- File locking for unmergeable files ("reserved checkouts").
- Path-based authorization.
- PHP, Python, Perl, and Java language bindings.
- Full MIME support - the MIME Type of each file can be viewed or changed, with the software knowing which MIME types can have their differences from previous versions shown.

For codes we will use Eclipse IDE. Because for server side and for Android (client side), Eclipse support us. So it is easy to compile and run projects in same time. For SVN, we will use Tortoise SVN tool to help us. About Eclipse; there is a little quote from Wikipedia. Eclipse is a multi-language software development platform comprising an IDE and a plug-in system to extend it. It is written primarily in Java and is used to develop applications in this language and, by means of the various plug-ins, in other languages as well—C/C++, Cobol, Python, Perl, PHP and more.

The initial codebase originated from Visual Age. In its default form it is meant for Java developers, consisting of the Java Development Tools (JDT). Users can extend its capabilities by installing plug-ins written for the Eclipse software framework, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Language packs provide translations into over a dozen natural languages.

Released under the terms of the Eclipse Public License, Eclipse is free and open source software.

For communicating we will use our web site, Google group, trac environment. To share some documents without code, we will use Google group.

3 CONFIGURATION MANAGEMENT STEPS

3.1 IDENTIFICATION

3.1.1 SOURCE CODE

Our source code will be always growing up. So it will be exposed to some changes. Controlling of the changes in project is so important due to possibility of causing crashes in project. So we are so careful to control and estimate results of the changes. So if any change request does not pass from the CCT, it will not survive. Java source codes are in two directories. One for server, one for client. Client side codes also have some APIs of the Android SDK.

3.1.2 WEB SIDE CODE

For now, we have no idea about web side codes. But most probably it will be PHP or Java. It will become absolute in few days.

3.1.3 DOCUMENTS

Most of the documents are in Google group. But some of these documents must be accessible from our project's web page. So they are in web pages directory. These documents are;

- Project proposal
- Requirements analysis report
- Initial design report
- Final design report
- Living schedule
- Configuration management plan.

There are other documents will be ready in following months. Important part of these documents are test cases, and documentation for Fizan. Living schedule will be changeable. According to our progress, it will be updated every week.

3.2 CONFIGURATION MANAGEMENT AND CONTROL

If any member of Macrohard believes project must change, who must reason of why project must change, which part affected from this change, which codes must be change. Who writes these and using trac informs CCT. If CCT okays it, project will be changed. After these, parts which will be changes participated to team members with respect to who coded which part. There will be some tests for the changes. If it will be desirable, changed version will be used, when any unwanted effect occurred, using SVN will be turned back to the old version.

3.3 CONFIGURATION STATUS ACCOUNTING

CSA is very important in CM because there will be crashes in Fizan development phase. So our group Macrohard will use some important and basic comments when updating repository files. We will be communicating when making changes.

3.4 REVIEWS

We talk in development of Fizan in possible times. Especially wednesdays, we make audits. In the spare times, after lessons, or when at home we wil talk about our project. So every detail of the project is known by every member of Macrohard. Web opportunities replace Fizan to our life.

4 PROJECT SCHEDULES AND CM MILESTONES

While improving our project, milestones will help us see how much time we have for a determined job to finish it and finally to finish the project. Also, being able to see upcoming tasks, in case of getting behind some of these milestones, we will be taking precautions not to delay other milestones or at least to make minimum changes to them. Followings are our milestones adjusted to demo dates and beta releases:

March 12th	:	User Interface of Android
March 16th	:	Database Implementation
March 16th	:	Communication Protocols
March 29th	:	Manipulation of Event/Friend
April 16th	:	Completion of Map
April 22nd	:	Gtalk Integration
May 31st	:	Integration of Modules
May 31st	:	Completion of website of Fizan
June 7th	:	Completion of tests for beta
June 11th	:	Final release
June 13th	:	Demo of the project and documentation

5 PROJECT RESOURCES

We have four main resources as below:

- **SVN:** Version controlling software provided by our department.
- **Trac:** Used for job assignments to group members, informing other group members about the process and Q&A between group members.
- **Google group:** Used for document construction and discussion for this process.
- **Web page:** To see current progress, living schedule, screenshots and project documentation

6 PLAN MAINTENANCE

CMP will be updated by configuration management update team when necessary. At our meetings, we will discuss the current situation of the project for needs of CMP and also living schedule will be updated according to development progress.