KONTRPİYE

CONFIGURATION MANAGEMENT PLAN

LINUX EVOLUTION SOCCER

1394865 Ufuk Dallı
1448505 Hakan Çağlar
1502145 Berker Batur
1502202 Uğur Büyükköy

Contents

1. Introduction 3
1.1 Purpose of Configuration Management Plan 3
1.2 Scope of Document 3
1.3 Definitions, Acronyms and Abbreviations 3
1.4 Document References 4
2. Management 4
2.1 Organization 4
2.1.1 Software Development Team 4
2.1.2 Testing Team 5
2.1.3 Release Control Team 5
2.1.4 Configuration Management Team 5
2.1.5 Configuration Control Team 5
2.2 SCM Responsibilities 5
2.3 Tools and Infrastructure 6
2.1.1 Eclipse IDE 6
2.1.1 SVN 6
3. CM Process 7
3 .1 Configuration Identification 7
3.1.1 Source Code 7
3.1.2 Database Management 7
3.1.3 Documentation 8
3.1.4 Additional Data 8
3.2 Configuration Management and Control 8
3.2.1 Software Change Request 8
3.2.2 Software Change Evaluation 9
3.2.3 Software Change Implementation 9
3.3 Configuration Status Accounting 10
3.4 Configuration Auditing 10
4. Project Schedule & CM Milestones 11
4.1 Schedule 11
4.2 Milestones 11
5. Project Resources 11
6. Plan Optimization 12

1.Introduction

1.1 Purpose of CMP

The purpose of this document is to define and explain the Configuration Management Plan for Linux Evolution Soccer. LES is a project in which many different modules work together and communicate to 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 as well because of this interaction. If not, the entities from apartly developed modules would not conform to each other which is not the intention. Configuration Management Plan discussed in this document will enable the concurrency and preempt this problem.

1.2 Scope of Document

This document was prepared by the developer team of LES, Kontrpiye. The scope is determination of a high level Configuration Management Plan for LES including the organization of the team, Configuration Management Process being used and the schedule of the project. Coping with versions mechanism, standardization, milestones, and resources are also in the scope of this document.

1.3 Definitions, Acronyms and Abbreviations

- √ **LES** Linux Evolution Soccer
- √ **SDT** Software Development Team
- √ **TT** Testing Team
- √ **RCT** Release Control Team
- √ **CMT** Configuration Management Team
- √ **CCT** Configuration Control Team
- √ **CM** Configuration Management

√ **CMP** Configuration Management Plan

√ **SVN** Subversion

√ **SCR** Software Change Request

√ **CSA** Configuration Status Accounting

1.4 Document References

√ Software Configuration Management lecture slides from Ceng492 Spring 2010 Website[1]

√ IEEE Standard for Software Configuration Management Plans – IEEE Std 828 – 1998[2]

2. Management

2.1 Organization

Kontrpiye consists of four people who are focused on different aspects of the development. The members are:

√ Berker Batur
√ Ufuk Dallı
√ Uğur Büyükköy
√ Hakan Çağlar

2.1.1 Software Development Team

Assigned Team Members: *ALL*

SDT is the team implementing all the modules in LES system. This team also integrates the implementations with the rest. Lastly, releasing a version is also a duty of this team.

[1] http://www.ceng.metu.edu.tr/courses/ceng490/

[2] http://standards.ieee.org/reading/ieee/std_public/description/se/1042-1987_desc.html

2.1.2 Testing Team

Assigned Team Members: *Berker Batur, Ufuk Dallı, Uğur Büyükköy*

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 SDT are in the duties of TT.

2.1.3 Release Control Team

Assigned Team Members: *Berker Batur*

RCT controls the current and next version, also gives feedback (SCR) to SDT about the current release.

2.1.4 Configuration Management Team

Assigned Team Members: *Ufuk Dallı, Hakan Çağlar*

The duty of this team is taking care of the CM organization by updating it every time when necessary.

2.1.5 Configuration Control Team

Assigned Team Members: *Berker Batur, Uğur Büyükköy*

This group has the duty of supervising the other four groups. Handling of SCRs are conducted within CCT.

2.2 SCM Responsibilities

Each member of Kontrpiye 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

The functions of the platforms made use of in the development are described below:

2.3.1 Eclipse IDE

Eclipse[3] is an open source integrated development environment. It is written in Java actually but very good and reliable for developing projects in many programming languages including C++ which is the only programming language used in the development of LES. Specifically, Eclipse CDT (Eclipse with C++ plug-in) is used in the development.

2.3.2 SVN

The version-control system is Subversion (SVN, renamed as Apache Subversion in February 2010). SVN helps us developers to backtrack, maintain current and old versions of implementations. Plus, since Kontrpiye consists of four members who work on the project at different times and places most of time, SVN will satisfy the mobility of the project on different operating systems (Linux and Windows) and hardware systems. Our department provides this tool and enough space for version controlling.

[3] http://www.eclipse.org/

3. CM Process

3.1 Configuration Identification

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. All of the implementation is done in C++ programming language.

Also some coding standardizations are determined in the development. This is a way of keeping metadata within the names of the variables. Actually in C++ standard library identifiers are mostly lowercase, but we prefer more upper case characters say instead of underscore ('_'). Our variable naming standards are:

√ Global Variables: *Globalvariable*

√ Local Variables: *locVar*

√ Variables with multiple keywords: *variableName*

3.1.2 Database Management

The whole data is stored in a single file which is actually an SQLite[4] database. The database is to be connected from the database module of LES and the relevant information data (player attributes etc) is to be transferred from the database into the ingame variables.

[4] http://www.sqlite.org/

3.1.3 Documentation

Documentation of LES has been done in several approaches. Those are:

√ Project Proposal Report
√ Requirement Analysis Report
√ Initial Design Report
√ Detailed Design Report
√ Revised Design Report
√ Configuration Management Report
√ Web Page sources
√ Comments on Source Codes
√ Developer Manual
√ User Manual

3.1.4 Additional Data

The SQLite database in LES keeps the records of the processable data. The other data files are kept in directories named correspondingly such as sound, image etc.

3.2 Configuration Management and Control

3.2.1 Software Change Request

There are two tools to make SCRs and handle them in the organization. First one is sort of informal and a project development portal of Zoho[5]. The second and the major one is trac in which every request is converted to a ticket and assigned to a member. A ticket can be updated in respects of requirements, postponed, reassigned to another member and when

[5] http://projects.zoho.com/portal/kontrpiye

assigned member is done with it, he closes the ticket. Other meta data kept within the trac are the following:

- √ Type of the SCR
- √ Time / Date information
- √ A keyword about the SCR
- √ A short summary and detailed information following about the SCR
- √ Tools, libraries to be involved to meet the SCR

3.2.2 Software Change Evaluation

Since Can Eroğul (Teaching Assistant of Kontrpiye) 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 an SCR is accepted during the meeting with the TA, a member is assigned to implement it till a due date. After the implementation is done the member uploads the file to the SVN repository and TT downloads the updated/added module and deploys it in their system to check if leads to any problem such as side-effects etc. Test cases are tried to cover all the aspects of the implementation and if it passes the test, nothing is wrong. Otherwise, the member implemented is informed and he is either re-assigned or another member takes over to debug the implementation.

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, he gets to inform all the other members and the TA as well about what he added or changed. Also he gets to log the reason of the change clearly and 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 Configuration Auditing

When a member has completed a task and is about to upload his source file to the SVN repository, he first needs to make sure that the code he wrote is working and also he should have tested it at least a bit to check if it works correct on trivial cases. After that TT takes over about that part since it is a well-known fact that nobody is likely to find his/her mistakes and somebody else needs to do it, TT in our case.

The members of Kontrpiye periodically check each others' work via the trac system and give feedback to each other personally when necessary. Project timeline determined is intended not to be changed if not absolutely necessary to follow the schedule. The most important auditing element is the exhaustive testing of LES before the final release to be conducted by the developer team with tough constraints not tested formerly.

4. Project Schedule & CM Milestones

4.1 Schedule

The living schedule with all tasks and milestones to be completed are in the website of Kontrpiye[6]. Good separation of components result in better functionalities, and also implementation of that particular component gets way easier with this paradigm. Having implemented the prototype at the end of the first term, implementation of better AI, new visual stuff, missing ingame functionalities, database and network module implementations are the major items in the list of the final release.

4.2 Milestones

√ **First Build** – 20 April

√ **Second Build** – 13 May

√ **Final Release** – 27 May

√ **Documentation** – 10 June

5. Project Resources

The website of Kontrpiye is where the user resources are stored. All of the documents produced during the development of the project are stored in here and also there is a news blog in the website which keeps the users informed.

The developer resources are deployed in the SVN installed on the departmant computers. We may also use Eclipse SVN to keep another model of LES in Eclipse's own Subversion utility. However, commits and check-out are done mainly on the departmental SVN account of Kontrpiye.

[6] http://senior.ceng.metu.edu.tr/2010/kontrpiye

6. Plan Optimization

Obeying the schedule is a must for the optimization issues on CMP. Obviously, delaying a milestone date might occur naturally once in a time but the intention is to keep this at minimum. Besides, since every milestone is supposed to be accomplished with the efforts of every single member, everyone is responsible from himself and has to report in to the rest of the team when he is asked for it. The demos presented to the TA in every two weeks is a big helper as well since official progress record is a bit of pushing and handy.