RAILWAY INFORMATION SYSTEM

CENG 492
Computer Engineering Design

Configuration Management Report

BUGBUSTERS

26.03.2010

Burkay SUCU

Caner ÇAKMAK

Halim Çağrı ATEŞ

Işıl Özge PEKEL

Contents

1. INTRODUCTION

1.1. Purpose of the Document

The purpose of this document is to identify and describe the configuration management (CM) process for *Railway Information System*. The intended audience for this document is BugBusters Team members and METU assistants and professors.

As BugBusters Team, we think that it is a vital task in software development process to prepare a configuration management plan and follow it throughout the development so that the changes and the overheads appearing during the implementation phases can be handled in an organized way.

1.2. Scope of the Document

The scope of this document is to clarify the milestones of the configuration management plan for the Railway Information Systems Project.

As a part of this plan, the organization of the team, the responsibilities of the team members and the configuration management issues will be discussed in detail. A schedule table for the assigned tasks will also be provided at the schedule part of this document.

1.3. Definitions, Acronyms and Abbreviations

RIS	Railway Information System
CM	Configuration Management
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
CMT	Configuration Management Team
SDT	Software Development Team
TT	Testing Team
CCT	Change Control Team

1.4. Overview

This document consists of six sections listed below:

1. **Introduction:**
   This section tells what this document is about. Abbreviations are also mentioned.

2. **Organization:**
   This section is about the team organization, team members' responsibilities, and

tools used for project management in that team structure.

3. **Process:**
   This section explains project development steps.

4. **Schedule :**
   Time plans of the project are described under this section.

5. **Resources :**
   This section underlines the needed resources.

6. **Plan Optimization:**
   In that section, it will be mentioned that in an unexpected situation how the plan will be adjusted to keep the project unaffected.

2. ORGANIZATION

2.1. Organization

The members of the *BugBusters* team will form 4 different sub-teams in order to manage the specific steps of the whole process. In each sub- team, there will be a team leader who is responsible for the quality of that sub-process. All tasks will be assigned by the team leader according to the experience and knowledge of the team members. In this way we aim to perfectly accomplish all tasks in each step. Since we have a limited time to develop the project, each team member will try to contribute to all steps.

2.1.1. Software Development Team

Software Development Team will be responsible for implementing the modules of the project and combining them with proper interfaces. They will release the partially implemented modules in the pre-determined times. If any bug or a missing part is found by TT during testing process, they will be handled as soon as possible. The leader of this team is Işıl.

2.1.2. Testing Team

Testing team will check if each implemented module corresponds to an entry in the requirements and design reports. They will give feedback to the SDT member who implemented that part. According to the type of the consistency, it will be decided whether implementing that part again or debugging is enough, or a change of request is necessary. The leader of this team is Caner.

2.1.3. Configuration Control Team

This team will be responsible for the management of other teams and will ensure a good communication between team members. In addition, it has the right to allow or reject a specific SCR if necessary. The cost of a change of request is taken into consideration elaborately in the decision process. If a SCR allowed, it monitors the implementation and integration of the new request. The leader of this team is Burkay.

2.1.4. Configuration Management Team

Configuration Management Team will be responsible for keeping the CMP up to date by caring maintenance of the CM organization. The leader of this team is Çağrı.

2.2. Responsibilities

Each member of BugBusters Team is a leader of a sub-team and responsible for the quality of his/her part. For the sake of team work, all implementations will be done in such a way that other team member can understand and improve it. Documentation and coding standards will be used for this purpose.

2.3. Tools and Infrastructure

2.3.1. SVN

Subversion is the new standard for version control and Software Configuration Management (SCM) for globally distributed organizations. Ease of use and out-of-the-box support for remote teams make Subversion the best solution for global projects, compared to legacy tools that are inadequate for distributed teams and too expensive to run [1].

All the members of BugBusters Team have installed the CollabNetSubversion[2], which is an enterprise-ready distribution of Subversion that includes certified binaries, platform-specific installers, and certified plugins, to load the source code of the project to the repository in the project SVN accounts. The directories in repository are created in a traditional way. “Trunk” directory will be used to keep the main source code. All modifications and integrations of new modules will be handled in this directory. In addition, each team member has his own “branch” directory to commit his/her current work which cannot be integrated to the main source code yet. After the completion of a branch work, the SDT will integrate it to the main source code. There is also a media directory to commit supplementary materials such as image, design, and configuration files.

2.3.2. NetBeans

Since we use J2EE to develop the RIS Project and we have divided the whole project into modules and we have decided to use NetBeans[3] platform, which allows fast and modular development for a J2EE project.

3. PROCESS

3.1. Identification

3.1.1. Source Code

The project's source code is kept in a central SVN repository provided by the department. Java is the main programming language used in the project. Since the main part of the project is web based, some web technologies like HTML and CSS are also used.

Especially for train monitoring the web pages are created as dynamic web pages and Java EE's JSP technology is used for this purpose.

NetBeans is decided to be the used as the IDE to develop the project. Also the NetBeans SVN plug-in will be used to connect central SVN repository and integrate the individual work done be the members to the main project.

3.1.2. Database

User data and the information about the stations and trains will be stored in a MySQL database. It will be accessed from the source code and also the system administrators will be able to access database directly or from an interface.

3.1.3. Documentation

Javadoc format will be used to generate API documentation of the source of the project. The resulting HTML files from Javadoc will be stored in the SVN directory hierarchy. The design reports are provided in the project website [4].

The user manuals will also be provided on the project website within the final package.

3.2. Management and Control

SVN and Trac will be used to configure and manage our Project RIS. As mentioned earlier, all team members will access the SVN repository and commit their modifications, additions or deletions. Via SVN this modifications will be available to all other members. But this is not enough for configuration management. With the use of Trac, we will be able to create a new ticket, assign it to one of the members and decide whether to implement it or not and eventually close the ticket.

3.2.1. Creating a New Ticket

If a change of request seems to be necessary, one of the members in TT will create a ticket in the Trac system. While creating this ticket, he has to attach necessary documents (like description or log) and determine the type of the ticket. In addition to that he has to assign it to a member for a period of time. As soon as a new ticket is created, an information mail is sent to all members.

3.2.2. Evaluating a New Ticket

Trac gives us the opportunity to edit the ticket, assign it to a new member or simply to reject it. As mentioned earlier, it is the responsibility of the CCT to review the new ticket and to accept or reject it. After the CCT considers the ticket in every aspect, rejection or approval decision is made. If the ticket is rejected, the status flag of the ticket is set accordingly. If not, then the assignee, deadlines, milestone and the tasks related to that ticket should be revised and ticket becomes active.

3.2.3. Implementing the Changes

The assignee has to fix the bug or implement the new request before the deadline and commit the changes to the repository. He also explains the modifications and closes the ticket. The TT adds new test cases to the system, tests the new implementation, and gives feedback to the members. The CCT is responsible for monitoring this process and ensures that all activities related to change of request accomplished appropriately.

3.3. Configuration Status Accounting

The living schedule provided in the project website shows the current status of the project to both project members and other people. People can see the progress of the tasks assigned to several team members. Besides SVN and Trac the website is a very easy and fast way of keeping track of the project for the members. This page will be updated regularly to keep the schedule up to date.

Also the weekly reports give an idea about the progress of the report. It mentions about the work done in that week and will be done in the next week. They are also provided in the project website.

3.4. Auditing

Auditing is a key part in developing our project. It is done in different levels.

For functional auditing, the first level is self auditing. Every member of the team must be sure that the code is working correctly before committing to SVN. That should be succeeded by compiling and testing the updated part.

Secondly, the TT should test the project and updated parts periodically to ensure the project working correctly. Also members can check others' work integrated with their parts to see if the modules are compatible with each other.

Lastly, the project should be tested in detail before every release. The TT must test the whole project as well as the modules part by part. Unit test and integration test should be done to make sure that the modules are well integrated. Regression test should be applied to see the how the project will be response in exceptional cases. Also process auditing should be done to obey the constraints of the project schedule.

4. PROJECT SCHEDULE

We divide all module implementations and project management into sub-tasks. So team members can work in parallel and development process becomes more effective. Every sub-task is assigned to some member/s and the progression is monitored from live schedule. Project live schedule may be followed from project website [4] under "Schedule" section. It is updated regularly and the progress may be followed accordingly. A table which shows the task list and people assigned to these tasks is below. The full live schedule table with dates and progress columns can be found on project website [4].

TASK	Sub-tasks	Assigned
RIS webpage design		Caner
	Design of the registration page	Caner
	Design of the machinist page	Caner
	Design of the home pages	Caner
	Design of the schedule page	Caner
	Design of knowledge-db page	Caner
Optimum path searching		Işıl
	Implementing the algorithm	Işıl
	Filtering the results	Işıl
	Integration to the interface	Işıl
Creating the map		Çağrı
	Deciding the station points	Çağrı
	Drawing the railways	Çağrı
	Map interaction implementations	Çağrı
	Marking coordinates on map	Çağrı
Loading city coordinates to database		Çağrı-Burkay
	Preparing data	Çağrı-Burkay
	Loading data to database tables	Çağrı-Burkay
Integration of search algorithm to new map		Işıl
Schedule sub-module implementation		Işıl
Knowledge database implementation		Işıl
	Sending a message to the system	Işıl
	Editing the message	Işıl
	Answering to that message	Işıl
	Escalation to other admins	Işıl
Design of journey module		Caner
Implementation of news module		Işıl
Design of news sub-module		Caner
Update sub-module		Işıl
Design of update sub-module		Caner
Accelerometer		Çağrı-Burkay
GPS Simulation		Burkay
	Parsing shp files	Burkay

	Parsing dbf files	Burkay
	Generating GPS data	Burkay
	Simulation program enhancing	Burkay
GPS and Map integration		Çağrı-Burkay
Accelerometer integration		Çağrı-Burkay
	Getting device working	Çağrı-Burkay
	Integrating the device to project	Çağrı-Burkay

4.1. Milestones

There are some strict deadlines which cannot to be changed and team members have to obey. These are:

06/04/2010	First Development Snapshot Demo
22/04/2010	Test Specification Report
13/05/2010	First Release Demo
10/06/2010	Final Package User Manual

5. RESOURCES

BugBusters resources for RIS project are:

- NetBeans IDE
- CollabNet Subversion
- Trac

Also Project website [4] will be used as a resource to publish reports, live schedule, and some news about project.

6. PLAN OPTIMIZATION

In order to meet the deadlines and complete the project on time, team members have to strictly obey the time constraints. In case there might be an unpredicted problem, the schedule will be adjusted and CM Plan will be changed accordingly.

REFERENCES

[1] *Collabnet subversion downloads*. (2010). Retrieved from http://www.open.collab.net/downloads/subversion/

[2] *The only Enterprise-ready distribution of subversion, certified by collabnet*. (2010). Retrieved from http://www.open.collab.net/products/subversion/

[3] *Welcome to netbeans*. (2010). Retrieved from http://netbeans.org/

[4] *Railway information system*. (2010). Retrieved from http://senior.ceng.metu.edu.tr/2010/bugbusters/

[5] *IEEE standard for software configuration management plans*. (1998). . Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720569&userType=inst