CONFIGURATION MANAGEMENT PLAN

PPTX TO HTML5 CONTENT CONVERTER

Ömer Baykal

Rustem Hashimov

Shamil Farajullayev

Nahid Hamidli

Table of Contents

1. Introduction

This document is the Configuration Management Plan of "PPTX to HTML5 Content Converter" project, created by the project team Limon.

1.1. Purpose of Configuration Management Plan

This Configuration Management Plan provides details of how the PPTX to HTML5 Content Converter project team, the team Limon, will manage the control of configuration items being developed under each module. Configuration Management procedures define how to record and process specification change requests, how to relate these to system modules and the methods used to identify different versions of the system. CM tools are used to store versions of system modules, build systems from those components and track the releases of system versions to customers. Hence, Configuration Management Plan allows software developers and software project managers to control and identify causes and results of changes in the software development process in order to implement these changes. Being guided by a CMP is very helpful for a software engineering team's members' communication and coordination.

1.2. Scope of the Document

The scope of this document is the determination of a high level Configuration Management Plan for PPTX to HTML5 Content Converter project. CMP presents configuration management activities that will be applied during the development process of the project and those activities explained in this document are applicable during the development and maintenance phases of the project.

Basically, this document consists of activities designed to manage changes and relate them. Coping with versioning mechanism, standardization, milestones, and an overview of utilized resources for the project are included, as well. The organization of the team and the schedule of the project are the other topics that will be covered in this document.

The related audience for this document is the project development team, and the faculty of Senior Computer Engineering Design (CENG492) course.

1.3. Definitions, Acronyms and Abbreviations

Definitions, acronyms and abbreviations that are used throughout this document are listed in the following table.

CM	Configuration Management
SDT	Software Development Team

SCR	Specification Change Request
CVS	Concurrent Version Systems
The Project	PPTX to HTML5 Content Converter Project
CMP	Configuration Management Plan
SCORM	Sharable Content Object Reference Module

1.4. Document References

[1]   The Project Website,

http://senior.ceng.metu.edu.tr/2012/limon

[2]   SCORM Standard,

http://en.wikipedia.org/wiki/Sharable_Content_Object_Reference_Model

[3]   Open XML SDK 2.0 in Microsoft Developer Network Page (MSDN),

http://msdn.microsoft.com/en-us/library/bb456488.aspx

[4]   Ribbon Class Documentation in MSDN,

http://msdn.microsoft.com/en-us/library/microsoft.office.tools.ribbon.aspx

1.5. Document Overview

This Configuration Management Plan document is divided into six sections, namely Introduction, The Organizations of CM Framework, Configuration Management Process, Project Schedule and CM Milestones, Project Resources and Plan Optimization. The content of those sections are as follows:

- ✓ **Introduction;** consists purpose and scope of the CM Plan Document, some definitions, acronyms and abbreviations used through the document, and references of the document.
- ✓ **The Organizations of CM Framework;** consists organization of the project team and responsibilities of its members. Moreover, explains tools and infrastructure of the project.
- ✓ **Configuration Management Process;** has the detailed information about the configuration management process of the project.
- ✓ **Project Schedule and CM Milestones;** clarifies the important dates for project development and the milestones set by the development team.
- ✓ **Project Resources;** contains the resources that is needed for completion of project development process.
- ✓ **Plan Optimization;** explains methods that will be used to optimize the Configuration Management.

2. The Organizations CM Framework

2.1. Organization

In work distribution, it is important for success to deliver the responsibilities to team members according to their previous experience and areas of interest. Limon the group consists of four computer engineers with different aspects. The members of the group are:

- ✓ Rustem Hashimov
- ✓ Shamil Farajullayev
- ✓ Ömer Baykal
- ✓ Nahid Hamidli

Each of the group members focuses on different aspect of the project.

2.1.1. Software Development Team

Software Development Team is responsible of implementing all modules in PPTX to HTML5 Content Converter Project. Moreover, this team integrates the implemented modules for final product. Creating the releases, project updates and bug fixes requested by Testing Team are also responsibilities of SDT.

2.1.2. Testing Team

Testing Team is responsible of testing the modules implemented by Software Development Team. The team will prepare test cases and use them to check whether the initial requirements of the modules are satisfied or not. According to test results, the team will give feedback to SDT.

2.1.3. Configuration Management Team

The Configuration Management Team is responsible of maintaining the CM organization of the project. So, this team can revise CM on necessary situations.

2.1.4. Configuration Control Team

The main responsibility of this team is supervising all the activities of other groups. Also, this group reviews Specification Change Requests; accepts, rejects and monitors the SCRs.

2.2. Responsibilities

Each member of the group is assigned to the organizational teams described in section 2.1 of this document according to their previous experience and areas of interest. The distribution of responsibilities of the members is as listed below.

Software Development Team: Ömer Baykal, Shamil Farajullayev, Rustem Hashimov, Nahid Hamidli

Testing Team: Ömer Baykal, Shamil Farajullayev, Rustem Hashimov, Nahid Hamidli

Configuration Management Team: Nahid Hamidli, Shamil Farajullayev

Configuration Control Team: Ömer Baykal, Rustem Hashimov

2.3. Tools and Infrastructure

2.3.1. Software Development

Microsoft Visual Studio: Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. Microsoft Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same IDE, which allows them to share tools and facilitates in the creation of mixed-language solutions. In addition, these languages leverage the functionality of the .NET Framework, which provides access to key technologies that simplify the development of applications.

2.3.2. Version Control

SVN: Apache Subversion (often abbreviated as SVN, after the command name svn) is a Software

Versioning and Revision Control System founded and sponsored in 2000 by CollabNet Inc. It is used by developers to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly-compatible successor to the widely used Concurrent Versions System (CVS). SVN maintains the current and old versions of source codes, web pages and

documentation on the server. We have configured TortoiseSVN to commit and update our source code since the project is being developed on Microsoft Windows platform.

2.3.3. Project Management

TRAC: TRAC is an open source, web-based project management and bug-tracking tool. The program is inspired by CVSTrac, and was originally named svntrac due to its ability to interface with SVN. It is developed and maintained by Edgewall Software. With the help of the TRAC, tasks will be assigned to the members of Software Development Team. Each member is going to use TRAC in order to show his progress.

2.3.4. Libraries

.NET Framework: The Microsoft .NET Framework is a software framework that runs primarily on Microsoft Windows. It includes a large library and supports several programming languages which allow language interoperability (each language can use code written in other languages). Programs written for the Microsoft .NET Framework executes in a software environment known as the Common Language Runtime (CLR), an application virtual machine that provides important services such as security, memory management, and exception handling. The class library and the CLR together constitute the .NET Framework. The .NET Framework's Base Class Library provides user interface, data access,

database connectivity, web application development, and network communications. The .NET Framework is intended to be used by most new applications created for the Windows platform. Microsoft .NET Framework also provides capability of creating plug-ins for Microsoft Office programs.

OpenXML SDK 2.0: Office Open XML [3] (also known as Open XML) is a zipped, XML-based file format developed by Microsoft for representing spreadsheets, charts, presentations and word processing documents. The Open XML SDK 2.0 for Microsoft Office, which is also supplied by Microsoft, simplifies the task of manipulating Open XML packages and the Open XML schema elements within a package. The classes in the Open XML SDK 2.0 encapsulate many common tasks (like getting elements) that developers perform on Open XML packages, so that they can perform complex operations with just a few lines of code.

3. Configuration Management Process

3.1. Identification

The identification of the items subject to the Configuration Management Process is crucial towards efficient construction and application of the plan. These items split into three categories, are given in the following subsections.

3.1.1. Source Code

3.1.1.1. User Interface Classes

- ✓ **Ribbon Class [4]:** This is the main class that constructs the user interfaces of the add-in that is designed to interact with users.
- ✓ **HierarchyEditor Class:** This class is responsible of constructing the Hierarchy Editor Pane User Interface that helps users to choose their desired hierarchy among the slides.

3.1.1.2. Library Classes

- ✓ **Publish Class:** This class has the duty of detecting slide constructions and elements, then creating corresponding HTMLs.
- ✓ **ImsXMLSerializer Class:** This class generates IMS Manifest XML file based on the slides contained in the presentation. This file is an essential and standardized document for SCORM [2].
- ✓ **HierarchyXMLSerializerDeserializer Class:** This class generates and parses the Hierarchy Tree XML that transfers hierarchy information between different modules of the project.

3.1.2. Data

There are three types of data objects used in the PPTX to HTML5 Content Converter Project.

3.1.2.1. PresentationML

PresentationML is the XML format used by Microsoft PowerPoint 2007 (and later versions) and that is part of the Office Open XML specification. It defines all the structure for the presentation related data.

3.1.2.2. Hierarchy Tree XML

Hierarchy Tree information will be kept in an XML file. Hierarchy Tree information will be used in two ways.
- ✓ Loading and modifying the hierarchy for a presentation.
- ✓ Reflecting the hierarchy in the Bookmark Module, that is designed to ease the navigation among created HTMLs.

3.1.2.3. IMS Manifest XML

IMS Manifest XML will also be placed in publish file directory. It will be used in order to create SCORM e-learning functionality.

3.1.3. Documentation

The PPTX to HTML5 Content Converter Project contains the following documentation:

- ✓ Project Development Documentation
    - ✓ Project Proposal
    - ✓ Software Requirements Specification Report
    - ✓ Initial Software Design Report
    - ✓ Detailed Software Design Report
    - ✓ Configuration Management Plan
    - ✓ Weekly Progress Reports
- ✓ User Manuals
    - ✓ Installation Manual
    - ✓ How-to-Use Manual
- ✓ Online Resources
    - ✓ Web Page

3.2. Configuration Management and Control

3.2.1. Specification Change Requests

Minor Specification Change Requests (SCR) are directly added to the system and handled by SVN and requires no extra information. However, major changes are controlled by the TRAC system. In this system, an SCR should consist of;
- ✓ Team member name
- ✓ Description
- ✓ Date
- ✓ Deadline

- ✓ Related Category/Module
- ✓ Priority
- ✓ Version number

When the request is made, a ticket is opened at TRAC and it can be seen by all members of the project team.

3.2.2. Specification Change Evaluation

The discussions about evaluation of Specification Change Requests are maintained on tickets in the tracking system. Some higher-level SCRs requiring a lot of change may be discussed during face-to-face meetings. However any decisions or key points on the SCR must be entered to the issue tracking system so as to keep the project organized. During the evaluation, each member can stress his opinions freely and the evaluation will be based on these opinions weighed against the goals of the project and the status in development.

3.2.3. Specification Change Implementation

If any Specification Change Request is approved after an evaluation, then the related Configuration Item will be updated, committed and upgraded to a new version through the SVN system.

3.3. Configuration Status Accounting

All of the configuration items have been written with details in sections 3.1 and 3.2 with a proper system to be able to handle the changes made to the software. Those changes ought to be stored somewhere so that it does not cause difficulties to find how the project changed for further development. Because of the necessity of accounting those configurations, we will keep pre-defined information with each status change. This will help the developing team to be easily integrated, goal oriented and be better individual developers since the status of the team have to be considered when a change is being made. In addition, aside from our team, people or organizations with interest to our project will be able to follow all the changes and be informed about the development process.

Each change commit will possess an identification given to that commission, such as a version number and should contain understandable and clear information about what changes have been made and in what module those changes have been placed.

While the development continues till the final project, we will show the information about the changes and updates made to the project in our SVN commit logs. Furthermore, there will be weekly meetings and progress reports, which aim to provide more detailed information about weekly statuses.

Because of the fact that we will show all the changes in SVN logs formally, those formal rules of each commission are defined as followed:

- ✓ Members' names worked on the update should be written
- ✓ Ticket related to this update should be given an identification (version) number
- ✓ Committed codes should be clarified with brief but clear information and comments so that everyone can understand even if he/she is not from the developing team
- ✓ There mustn't be any conflicting updates, or more than one update in each commission. Therefore, before committing, developers do check the recent state of the project

More information about the project can be found at our project website [1]. Weekly iterations and snapshot demos will be shown in the news section of the webpage. In addition, one can find contact information of each member of the developing team on the website.

3.4.Auditing

All members of the project team are responsible for auditing the configuration process. Those changes should be checked finally weather it fits in the latest state of the software without creating any conflicts with any parts of the program. Aside from weekly meetings, developers arrange a meeting to combine the progress made in latest iteration, and check for any possible errors created by the integration process of different modules. It is preferred that before a team member commits the update he made, he consults the other members of the team so that conflicts are avoided at the beginning. SVN repository must not contain any conflicting updates, since the reverse process is much more difficult so auditing must be considered seriously by each team member. Aside from compiling the recent code on just one computer, it is recommended to compile and check the working status of the code on different machines for the sake of compatibility. Finally, team members gather weekly to combine the individual reports about the weekly process and write a general progress report to be presented to the responsible project assistants.

4. Project Schedules and CM Milestones

4.1. Project Schedules

We are following the Gantt chart that was presented in the Detailed Design Document of the project. Furthermore, we assign each member a task every 2 weeks, which are selected carefully to maximize the development pace of the software. First part of the Gantt chart has been completed, a few of the second part is finished while some are still in progress. Iteration Schedule provides more detailed process about the project with clearly separated tasks which are preassigned to members evenly.

4.2. CM Milestones

We have defined number of milestones to be able to track the development process more easily. After each iteration period, which is 2 weeks, we present a snapshot demo of the current state of the program, which clearly summarizes what we have done so far. Milestones of the project, shown with progress details, are demonstrated below:

Milestone	Status	Due Date
Initial Software Kit and Demo Conversion	Done	20.01.2012
General Output Layout	Done	11.03.2012
Bookmark Panel Development	Done	23.03.2012
Constructing Main Hierarchy Features	Done	25.03.2012
Main Text and Image Conversion	In Progress	08.04.2012
SCORM Application and Monitoring	In Progress	10.05.2012
Touch Gestures	Waiting	06.05.2012
Final SCORM Package	Waiting	20.05.2012
Final Tests and Debugging	Waiting	09.06.2012

5. Project Resources

5.1. Software Tools

The resources of the project can be listed as follows:

- ✓ SVN; Version Control System
- ✓ TRAC; Issue Tracking System
- ✓ Microsoft Visual Studio; Integrated Development Environment
- ✓ .NET Framework, OpenXML SDK 2.0; Libraries

6. Plan Optimization

During software development process of the PPTX to HTML5 Content Converter Project, it would be important to obey CMP for optimization of the project management. All the tasks are defined in the living schedule and all members of the team will follow these tasks. On the other hand, team may need to make some changes in the deadlines of the living schedule during the development process. When an update or change occurs in the schedule, it will be reflected to the living schedule of the project. All those changes and current version of the schedule will be able to be tracked using the project website. Also, if some optimizations to the project are agreed in the project group meetings, they will be clarified on the TRAC system.