**PNW University**

**Website and Course work portal**

**Configuration Management Plan**

**April 26, 2017**

**AAU Gold Team**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 02/27/2017 | 1.0 | Document creation | Jacob Hertl, Aaron Tonkovich |
| 04/24/2017 | 2.0 | Document Update | Jacob Hertl, Aaron Tonkovich |
| 04/25/2017 | 2.0 | Final Document Approval | Philip Parker, PM |

Table of Contents

[1.0 Introduction 3](#_Toc474312711)

[1.1 Purpose 3](#_Toc474312712)

[1.2 Scope 3](#_Toc474312713)

[2.0 Configuration Management 4](#_Toc474312714)

[2.1 Organization 4](#_Toc474312715)

[2.2 Responsibilities 4](#_Toc474312716)

[2.3 Applicable Policies, Directives, and Procedures 5](#_Toc474312717)

[2.4 Management of the SCM Process 5](#_Toc474312718)

[3.0 Configuration Management Activities 5](#_Toc474312719)

[3.1 Configuration Identification 6](#_Toc474312720)

[3.1.1 Identifying Configuration Items 6](#_Toc474312721)

[3.1.2 Naming Configuration items 7](#_Toc474312722)

[3.1.3 Acquiring configuration Items 7](#_Toc474312723)

[3.2 Configuration Control 7](#_Toc474312724)

[3.2.1 Requesting Changes 8](#_Toc474312725)

[3.2.2 Evaluating Changes 8](#_Toc474312726)

[3.2.3 Approving or Disapproving Changes 9](#_Toc474312727)

[3.2.4 Implementing Changes 9](#_Toc474312728)

[3.3 Configuration Status Accounting 9](#_Toc474312729)

[3.4 Configuration Evaluation and Reviews 9](#_Toc474312730)

[3.5 Interface Control 10](#_Toc474312731)

[3.6 Subcontractor / vendor control 10](#_Toc474312732)

[3.7 Release management and delivery 10](#_Toc474312733)

[4.0 Configuration Management Schedules 10](#_Toc474312734)

[5.0 Configuration Management Resources 11](#_Toc474312735)

[6.0 Configuration Management Plan Maintenance 11](#_Toc474312736)

**THIS IS A WORKING DOCUMENT THAT WILL BE UPDATED AS THE PROJECT**

**PROGRESSES. COMMENTS RECEIVED FROM THE CITSS PROJECT TEAM, CUSTOMERS, AND STAKEHOLDERS SHOULD ONLY BE CIRCULATED AFTER COORDINATION WITH THE PROJECT MANAGER.**

# 1.0 Introduction

## 1.1 Purpose

The Configuration Management Plan (CMP) defines the methods that will be used to manage changes to the Above Average University Website(AAUW) production and test bed environments. AAUW will be implemented in major version releases, with each release having additional features. The AAUW software architecture is composed of. The hardware architecture is a database, website server, client, and the network.

## 1.2 Scope

Above Average University needs an informative website that can also provide access to student and faculty member portals where grades and course work can be accessed amongst other features. AAUW will be the center for all university online activity, so it is imperative that it is speedy and reliable. From their respective portal there will be a suite of features to provide the basic functionality to allow the university to operate.

The major functional objectives of AAUW are:

* Speed and ease of use of the entire website
* Provide and informative connective main site
* Login that recognizes user type
* Adaptive database to allow the creation and maintenance of courses
* Search function for courses
* Manageable course work web application

# 

# 2.0 Configuration Management

## 2.1 Organization

The following table identifies all personnel and/or groups who have responsibilities with regard to configuration management for AAUW.

|  |  |
| --- | --- |
| **Role** | **Name** |
| Integration Engineer /Coder | Jacob Hertl,  Aaron Tonkovich |
| Project Manager /Coder | Phil Parker |
| Software Architect /Coder | Casey Boatman,  Justin Andras |
| Requirement Engineer /Coder | Ryan Peters |
| Testing Engineer /Coder | Jonathon Nowak,  Joshua Potrawski |

## 2.2 Responsibilities

Each of the roles identified in the earlier table have specific configuration management responsibilities. The responsibilities are defined below.

##### Integration Engineer

* Maintain SCMP standards and protocols
* Manage Software Configurations
* Manage Source code changes, merges and branch structure
* Review commits to SCM and evaluate them

##### Project Manager

* Manages all team members
* Help Integration engineers uphold SCMP
* Review commits to SCM and evaluate them

##### Programmers

* Adhere to standards presented in this document
* Help review commits to SCM and evaluate them

## 2.3 Applicable Policies, Directives, and Procedures

We will follow all of the IEEE’s standards for software configuration management.

## 2.4 Management of the SCM Process

## 

The integration engineers are responsible for the management of the SCM process as well as acquiring, maintaining, and backing up the configuration items.

# 3.0 Configuration Management Activities

SCM activities will be primarily carried out by the integration engineers but all team members may participate. These activities include but are not limited to:

* Reviewing commits
* Manually selecting conflicting merges
* Reviewing branch structure

## 3.1 Configuration Identification

There are three major AAUW configuration areas:

* Project Documentation
* COTS Software
* Hardware

Specific items in each of these areas may be added or deleted during the AAUW life cycle. This document will be modified to reflect any changes.

### 3.1.1 Identifying Configuration Items

The project leader and other specified members are responsible for identifying all configuration items. All configuration items will be located on the shared GitHub repository.

The information below is required to be updated every time the document is delivered. The version number and revision date will help track changes. The document name and publication date will be used to identify the document.

* Document Name
* Publication Date
* Revision Date
* Version Number
* Author(s)

### 3.1.2 Naming Configuration items

The configuration leader, being all engineers of the project, have the responsibility for labeling all CI’s and will follow a set of conventions.

### 3.1.3 Acquiring configuration Items

Configuration items are available to all engineers via Git. All members of the project can acquire all these configuration items by using Git to download them from our GitHub repository.

## 3.2 Configuration Control

This section spells out the process whereby configuration items are changed. This process should be flexible enough to allow quick changes, but controlled enough to keep changes very orderly.

### 

### 3.2.1 Requesting Changes

Any changes to the source code will submitted to the AAUW GitHub repository via commit of well organized, commented code with a detailed commit comment.

### 3.2.2 Evaluating Changes

The Project Manager and integration engineers will regularly check our GitHub repository for recent commit and will review them (all programmers will assist with this as well). They will look over the changes for obvious errors and to see if the changes are applicable. If the commit contains errors or is incomplete in any way, the reviewer will contact the submitter so they can revise it. If the commit is good, the PM or an integration engineer will approve it and make an impact assessment of the change. The following is a list of impact definitions.

**Emergency -** If the change is not made soon the entire website may not function or the change may be needed to restore operation to the site.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Critical -** |  | The impact of not making the change would significantly impact website, but would not cease it operation. |
| **Routine -** |  | A normal change request. This will be dealt with according to the current schedule or plan. |
| **Deferred -** |  | A non-important change request, it is beneficial to the site, but is delayed because of other project schedules or tasks take priority. |

The impact assessment will give the integration engineers a gauge to measure the importance of the changes. The integration engineers will oversee the merging in to the repository and help resolve conflicts.

### 3.2.3 Approving or Disapproving Changes

### 

We will use a peer review system in which at least 3 other programmers must review a commit before it is merged. If at any time the change is found to be unacceptable the change will have to be fixed according to the reviewer’s comment and resubmitted for review again.

### 3.2.4 Implementing Changes

Once the change has been approved by a designated member a pull request on GitHub will also be issued and approved. The changes located in the separate branch will then be merged into the master branch and the all members can then review the changes.

## 3.3 Configuration Status Accounting

Each configuration item will be tracked in great detail automatically via Git. GitHub will also be used for further analytics. Git will track the version, date, status, and the changes of each configuration item. GitHub tracks everything that is submitted and information about it. This information includes but is not limited to: Author, comments, description, time, issues, and where in the branch structure it was submitted. In addition to this it can keep track of documentation, contributions, traffic, and pull requests.

## 3.4 Configuration Evaluation and Reviews

Auditing verifies that configuration items reflect operational objectives, meet DOE standards, and satisfy customer requirements. The team will follow a technical review process. This will include Structured Walkthroughs, In-Stage Assessments, Stage Exits, and System Acceptance.

Each of these processes are described below.

**Structured Walkthroughs** are informal meetings among the project team to review and evaluate technical aspects of the application.

**In-Stage Assessments** will be conducted with a Quality Assurance (QA) consultant near the end of each project stage. This is a technical review to assure that the established project management processes are being followed effectively.

**Stage Exits** are formal meetings with a group of selected individuals to review and evaluate the current status of the project. When a stage has been successfully "exited", it indicates that all deliverables due to date have been completed, all outstanding issues have an acceptable action plan, and there is a sound plan for the remainder of the project. All approvers must provide a written position of concur/non-concur at the Stage Exit.

**System Acceptance** provides a formal basis for determining whether the system is fully operational and has satisfied customer requirements.

## 

## 3.5 Interface Control

The interface will be managed by all engineers of the project.

## 3.6 Subcontractor / vendor control

Due to the small scope of this project there will not be a need for vendors or subcontractors.

## 3.7 Release management and delivery

Internal Releases will be made often and can be obtained from our GitHub repository. The final version will be delivered on a DVD disk burned from a synced local repository.

# 4.0 Configuration Management Schedules

The integration engineers will check all commits daily and perform any necessary maintenance weekly. All team members will be encouraged to peer review others commits daily, but are not required to do so.

# 5.0 Configuration Management Resources

As of now we are using Git to enforce Source control. Git will automatically merge branches and changes to a certain degree but may require an integration engineer or the submitter to manually select changes. GitHub will be our primary means of storing our repository and allowing all team members to connect to it.

# 6.0 Configuration Management Plan Maintenance

The integration engineers will maintain this plan and may be aided by the project manager. The\is plan and Software Configuration Management will also be guided by the AAUW Software Quality Assurance Plan (SQAP). Minor changes made to this plan will be done throughout the project lifecycle and done so by an update. Any significant changes will result in a new version release of this plan.