Team 16

NGO Information Management Suite 1.0

Software Configuration Management Plan

Overview:

Configuration Management (CM) Plan document describes how CM of the project will be carried out.

Intended Audience:

The intended audience of the our CM Plan is the project manager, project team, project sponsor and the NGO whose support is needed to carry out configuration plans.

Revision History:

Version	Primary Author(s)	Reviewed By	Date Completed
1.0	Surbhi Jatan Jesal	Aakash	2nd April 2012

Table of Contents

1 INTRODUCTION

The overall objective of a Configuration Management (CM) Plan is to document about CM of the project, what CM tools will be used, and how they will be applied by the project to promote success. The NIMS Software Configuration Management Plan documents methods to be used for the identification of software items, control and implementation of change, and recording and reporting change implementation status. The NIMS CM Plan defines the project’s structure and methods for

- Identifying, defining, and base lining CM activities.
- Controlling modifications and releases of CM activities.
- Reporting and recording status of CM activities and any requested modifications
- Ensuring completeness, consistency, and correctness of CM activities.
- Controlling storage, handling, and delivery of the CM activities.

2 REFERENCE DOCUMENTS

- IEEE Std 1042-1987
- IEEE Std 828^{TM}-2005
- http://git-scm.com

3 MANAGEMENT

This section provides information describing the allocation of responsibilities and authorities for software configuration management activities to organizations and individuals within the project structure.

3.1 ORGANIZATION

The main in Configuration Management is given to Configuration Manager. This role is being held by Aakash Solanki.

3.2 RESPONSIBILITES

The main roles and responsibilities distributed among the team members for configuration management are:

Roles	Responsibilities	Member or group associated
Project manager	The Project Manager has knowledge of the state and content of all documents and follows them up. If for any reason configuration manager is not present he assigns his role to someone else from the team to perform on his behalf.	Aakash Solanki
Configuration manager	Organizes software configuration management. Configuration Manager is also responsible for installation &maintenance of configuration management tool and providing support & training to its users.	Aakash Solanki
Developers	All the members of team 16 come under this role and they follow the plan for the configuration management using configuration management tool.	Team 16

3.3 APPLICABLE, POLICIES AND DIRECTIVES

- All relevant products are to be added and stored to the GIT repository. Documents (not on the repository but on the mailing lists), website files, source-code should immediately be available to the developers.
- Committed source-code should always be tested and must not contain any simple errors. This rule can exceptionally be overruled with explicit permission of the Implementation Leader (Aakash Solanki) to let others help resolve errors.
- Revision conflicts are to be resolved by the developers themselves as soon as possible. Communication with other developers is advised.

4 ACTIVITIES

This section of the Software Configuration Management Plan deals with the activities that are undertaken to manage the configuration of the software system as well as the documentation produced as a result of the project.

4.1 CONFIGURATION IDENTIFICATION

Configuration Identification deals with identifying and classifying development objects into:

- Controlled – objects that are to be placed under configuration control
- Pre-controlled – objects that are not yet but will be under configuration control
- Uncontrolled – objects that are not and never will be under configuration control

4.1.1 Identifying Configuration Objects

Configuration Object	Configuration State
Feasibility Study Report	Controlled
Project Plan	Controlled
System requirement Specification Document	Controlled
High Level Design Document	Controlled
Low Level Design Document	Controlled
Test Plan	Pre-Controlled
Test Cases	Pre-Controlled
Source Code	Controlled
User Manual	Controlled

4.1.2 Naming configuration Objects

Git identifies each revision with a unique SHA1 id, which is a long 160-bit number code written in hexadecimal. The latest revision can be referred to by HEAD, its parent by HEAD^, and its parent to be HEAD^^ and so on.

The names given to the configuration objects have to be as short and as meaningful as possible. Spaces in names are replaced by underscores.

4.1.3 Acquiring Configuration Objects

Configuration objects can be acquired by using the commit and checkout procedures only. The repository can be accessed:

Locally – C:\\wamp\www\

Remote (Online) - https://bitbucket.org/SkishChampi/nims-s/

4.2 CONFIGURATION CONTROL

Configuration control deals with the management of changes that are made to the controlled objects. These changes may be due to either error correction or object enhancement. Configuration control involves managing and regulating all the requests, evaluations, approvals or disapprovals and implementations of the changes made to the objects. This section also involves the records to be used for tracking and documenting the sequence of steps for each change.

4.2.1 Requesting Changes

After changes are made to a configuration object, the Git add and commit procedures are used to push the changed configuration object on to the master repository. After this is done, a pull request is created by the team member requesting the changes.

4.2.2 Evaluating Changes

Changes made to a configuration object are evaluated by the review team. After evaluation, the pull request is either accepted or rejected based on the evaluation results.

4.2.3 Implementing Changes

Once a pull request is accepted, Git stores the new, changed configuration object in the repository along with the changes that were made from the previous version. Then it deletes the older version from the repository.

5 RESOURCES

- Bitbucket is a hosting site for the distributed version control systems (DVCS) Git.
- Bitbucket is an up and coming free hosting site for either private and public git or mercurial repositories. Free features include private repository (up to 5 users, unlimited educational) and public repository (unlimited open-source). Bitbucket also has Wiki and issue tracking features.
- Git is a free & open source, distributed version control system designed to handle everything from small to very large projects with speed and efficiency.
- Every Git clone is a full-fledged repository with complete history and full revision tracking capabilities, not dependent on network access or a central server.

Branching and merging are fast and easy to do.

- Git is used for version control of files, much like tools such as Mercurial, Bazaar, Subversion, CVS, Perforce, and Team Foundation Server.

6 PLAN MAINTENANCE

The Configuration Manager is responsible for monitoring the Software Configuration Management Plan. The Software Configuration Management Plan has to be updated on the introduction of new Software Configuration Management guidelines or the modification of the old guidelines. Changes made to the Software Configuration Management Plan are evaluated and approved by the review team. When the review team reviews and modifies the Software Configuration Management Plan, the changes made are communicated to all the team members along with the new, modified Software Configuration Management Plan.