**Agile Delivery Services BPA**

**RFQ ID: RFQ993471**

**Pool 2**

**Evidence Item No.: g**

**CONFIGURATION MANAGEMENT PLAN**

**Version Number: 1.1**

**Version Date: July 7, 2015**

VERSION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Number** | **Implemented By** | **Revision Date** | **Description of Change** |
| 1.0 | Parvez Aziz | 06/20/2015 | Initial Version |
| 1.1 | Parvez Aziz | 07/05/2015 | Update Source Tree Setup section |

TABLE OF CONTENTS

[1. Introduction 4](#_Toc423951971)

[1.1 Purpose 4](#_Toc423951972)

[2. Configuration Management 4](#_Toc423951973)

[2.1 Planning 4](#_Toc423951974)

[2.2 Staff](#_Toc423951975)

[2.3 Tools 5](#_Toc423951976)

[3. Configuration Management Activities 5](#_Toc423951977)

[3.1 Configuration Items 5](#_Toc423951978)

[3.2 CONFIGURATION IDENTIFICATION 5](#_Toc423951979)

[3.2.1 Labeling 5](#_Toc423951980)

[3.2.2 Naming Conventions 6](#_Toc423951981)

[3.3 Repository Folders 6](#_Toc423951982)

[3.4 Source Tree Setup 6](#_Toc423951983)

# Introduction

**INCATech** is responsible for the development and deployment of software (SW) necessary to support the functions of the General Services Administration (GSA). The CM process provides the framework for the identification, control, status accounting, roles, responsibilities, standards, and procedures for developing and implementing the change management and control processes. To elaborate, the CM process strives for the following:

* Providing a structural framework of administrative functions and processes to support development and maintenance efforts
* Establishing organizational standards for identifying, controlling, tracking,   
  auditing, and documenting development and implementation of changes to controlled configuration items (CI) of a system or its components
* Defining the standards for establishing approvals for all changes to a system or its components, and supporting CIs

## 1.1 Purpose

CM is responsible for managing the configuration of all SW and related documents, developed or tracked for GSA. The CM Plan is designed to:

* Direct users to the appropriate subordinate document that details the interactions and their outcomes as it relates to the SW change process.
* Encompass the lifecycle of all GSA systems and its components.
* Identify requirements for CM status accounting.
* Define roles and responsibilities.
* Define configuration item identification requirements.

# Configuration Management

## 2.1 Planning

During the course of the Software Development Lifecycle (SDLC), staff and tools are identified to support all activities throughout the lifecycle of the system or its component.

## 2.2 Staff

CM staff will support all CM activities defined in this document. Each of the areas defined in section 3 – responsibilities will have staff dedicated to those tasks.

**Note**: GSA CM staff is available 8:00 am to 5:00 pm EST to attend any meetings as required by GSA management to support the activities of the program and the requirements of GSA

|  |  |
| --- | --- |
| **Role** | **Name** |
| Project Manager | Brian Skapura (INCATech) |
| Configuration Manager | David Alicea (INCATech) |

## 

## 2.3 Tools

CM staff uses the following tools:

* **Window’s Command Window** used to access GitHub folders/Files.
* **Office 365 SharePoint** is used to collaborate on writing and sharing content. It can be just words, or you can add images, diagrams, activity stream and more.
* **Source Tree** is a repository for codes resides in your local machine.
* **GitHub** is a web-based hosting service for software development projects that use the Git revision control system. <https://github.com/Incatech-Corp/gsa-agile>
* **Win Merge** is a tool used to compare Source Tree and GitHub source code changes.

# Configuration Management Activities

## 3.1 Configuration Items

A system is a group of interrelated, interacting, or interdependent supporting components. Multiple applications and standalone systems that have limited/no interaction with other systems and/or SW make up the complex systems that GSA maintains. Components can be identified as the following:

* **Support System/Reusable**: Code developed to meet a specific business need and can be shared for common functionality
* **Utility**: Code developed to assist in the collection and presentation of pertinent data
* **Database**: Information repository that stores information accessible to one or more GSA Applications
* **System software**: Serving a mission for GSA that runs independently and serves with one or more business needs. Systems are developed by utilizing a formal development program

## 3.2 CONFIGURATION IDENTIFICATION

### 3.2.1 Labeling

All CIs will be delivered to GSA via CM. Delivery of CIs directly from Development groups to GSA is NOT permitted.

* Each production baseline/CI being submitted to GSA CM must be checked into GitHub and labeled by the Development group. The label naming convention will be as follows: GSA\_01.00.00.1, where GSA represents the app name/acronym, 01.00.00 represents the version number, and .1 represents the iteration of the submission. The iteration number will sequentially increase with each submission to CM.
* The submission notification to CM will be an e-mail that includes the following information:
* Application Name
* Version Number
* Label applied
* Action requested
* The development group will submit all CIs to GSA CM no later than two business days prior to the scheduled Release Planning meeting. Failing to submit all CIs per this requirement may cause delay of the Release.

### 3.2.2 Naming Conventions

Each release has a version number that consists of four levels, with each level separated by a period (nn.mm.pp.od): GSA.01.00.00

* **nn** – The first level is used for new releases, or increases for major release type changes: 01.00.00
* **mm** – The second level increases for minor release type changes: 01.01.00
* **pp** – The third level increases for patch release type changes: 01.01.01
* **od** – The fourth level increases for operational data type changes: 01.01.01.01

## 3.3 Repository Folders

**Source Tree and Local PC Folder Structure**

Create the folder directory structure on the local C drive of the Source Tree machine.

**Note**: The attached ZIP file to a local drive and extract to create the directory structure.

**SharePoint Folder Structure**

Create the folder directory structure under the Releases folder in the SharePoint Configuration Management sub site to hold the documentation deliverables.

Release Folder, {GSA\_.NN.MM.PP.DD}.

## 3.4 Source Tree Setup

Download SourceTreeSetup\_1.4.1 from Tools folder

Install Source tree into default location.

During installation:

* Enter your full name and email address (your @gsoft.com email, if you work for GSoft)
* Choose the option to install the self-contained Git version to be used by when prompted
* Skip the Mercurial option
* Choose Putty as SSH option
* Enter your github.com username and password

Checkout the GSA Agile project from GitHub: <https://github.com/Incatech-Corp/gsa-agile>