Plan de Gestión de Configuración

Versión 1.4

**HACHIKO**

# Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 25/05/2019 | 1.1 | Planificación de SCM - Propósito | Ruesta Sedano Alexander |
| 26/05/2019 | 1.2 | Roles, Responsabilidad y Cantidad | Sanchez EsparzaGerson |
| 27/05/2019 | 1.3 | Políticas, Directrices y Procedimientos | Satornicio Medina Andres |
| 28/05/2019 | 1.4 | Herramientas, Entorno e Infraestructura | Aguirre Olazo Estéfano |

Contenido

[Historial de Revisiones 2](#_Toc7692295)

[1. Planificación de la SCM 4](#_Toc7692296)

[1.1 Propósito 4](#_Toc7692297)

[1.2 Roles, Responsabilidades y Cantidad 4](#_Toc7692298)

[1.3 Políticas, Directrices y Procedimientos 6](#_Toc7692299)

[1.4 Herramientas, Entorno e Infraestructura 6](#_Toc7692300)

[1.4.1 Herramientas 6](#_Toc7692301)

[1.4.2 Entorno 8](#_Toc7692302)

[1.4.3 Infraestructura 8](#_Toc7692303)

# Planificación de la SCM

## Propósito

Actualmente contamos con 7 proyectos de los cuales 3 están aún en fase de desarrollo como SAV, mientras los otros 4 en producción. Nuestra consultora cuenta con diversos productos de software, que durante si desarrollo, dieron muchos problemas con los cambios en las versiones y al restaurar versiones funcionales. Este problema lo relacionamos al hecho de no contar con una guía o documento que estandarice la gestión de versiones.

Se espera la aplicación de este plan a todos los proyectos existentes en sus diversas etapas, sin importar su alcance. Se aplicarán las actividades a lo largo del ciclo de vida del software y su desarrollo para garantizar la calidad del producto.

Como finalidad este plan de gestión de la configuración debe garantizar que no se realicen cambios no controlados y que todo el equipo participante del desarrollo acceda a la versión adecuada del producto que utiliza en todo el ciclo de vida del desarrollo de software. Con esto podemos obtener mejor calidad que conlleva a una satisfacción de nuestros clientes y de una mejor a nuestra consultora.

## Roles, Responsabilidades y Cantidad

A continuación, se presenta en la Tabla 1 los roles y sus responsabilidades correspondientes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Roles** | **Responsabilidad** | **Cantidad** | **Autoridad** |
| Gestor de la gestión de la configuración | * Encargado de implementar, mantener y mejorar la gestión de configuración. * Responsable de la utilización de herramientas durante la gestión de configuración. | 1 | El primer responsable del cumplimiento y del proceso de gestión de configuración. |
| Comité de control de cambios | * Evaluación de registro de eventos. * Gestión de solicitudes de cambios relevantes * Seguimiento de registro de eventos y solicitudes de cambios a través de sus respectivos ciclos de vida. * Proveer apoyo a la persona que registró el evento. * Coordinar con otros Comité de Control de Cambios. * Coordinar con la gestión de proyecto u otra gestión relevante. | 1 | Evalúa cualquier tipo de modificación |
| Equipo de desarrollo | * Participa de los requerimientos. * Aborda diversos aspectos del desarrollo, incluidos documentos, código, entrenamiento, COTS y hardware. | 3 – 5 | Depende de cada miembro y sus obligaciones. |
| Bibliotecarios | * Define y da mantenimiento a las bibliotecas que son usadas durante la gestión de configuración. * Es el encargado de asegurarse que los aspectos prácticos de la gestión de configuración trabajen entre sí adecuadamente. | 1 | Controla el ingreso a líneas base garantizando el uso de los procedimientos formales definidos en el plan de SCM. |

**Tabla 1. Roles, Responsabilidades y Cantidad**

## 1.3 Políticas, Directrices y Procedimientos

Los procedimientos para modificar los archivos que pertenecen al repositorio son los siguientes:

* Debe indicarse a todos los miembros que documento está a punto de ser editado, con el objetivo de evitar que un ítem se sobrescriba, o se haga el trabajo dos veces.
* Se modifica el ítem en la rama asignada al integrante
* Se suben los cambios a su rama correspondiente.
* Se hacen los merge necesarios para que la modificación del ítem se encuentre disponible para todos los integrantes.
* Hay que informar que el cambio fue realizado.

Los procedimientos para modificar los archivos que pertenecen al repositorio son los siguientes:

* Debe indicarse a todos los miembros que documento está a punto de ser editado, con el objetivo de evitar que un ítem se sobrescriba, o se haga el trabajo dos veces.
* Se modifica el ítem en la rama asignada al integrante
* Se suben los cambios a su rama correspondiente.
* Se hacen los merge necesarios para que la modificación del ítem se encuentre disponible para todos los integrantes.
* Hay que informar que el cambio fue realizado.

Todos los integrantes tienen derechos de administrador: permisos de escritura, de lectura y de creación de documentos.

## 1.4 Herramientas, Entorno e Infraestructura

### 1.4.1 Herramientas

**GIT**

Durante el proceso de gestión de configuración se utilizará la herramienta git para el control de versiones del producto. Su propósito es llevar registros de los cambios y coordinar el trabajo que varias personas realizan sobre archivos compartidos. En la figura 1 se muestra la estructura que se utilizará.



**Figura 1. Estructura GIT**

Para realizar el seguimiento de nuestro trabajo, se define los distintos estados en que pueden estar nuestros ficheros en Git.

**Sin seguimiento**: Se encuentran en ese estado todos los ficheros que han sido creados fuera de Git, y nunca los hemos incorporado al gestor de versiones.

**Sin modificar:** Se trata de la situación base en la que se encuentran todos los ficheros sujetos al gestor de versiones, y sobre los que nunca se ha hecho nada.

**Modificado:** Todos aquellos ficheros que están bajo el control de versiones, y sobre los que hemos realizado alguna modificación.

**Preparados:** Son archivos que hemos modificado, y consideramos que ya están listos para entregar, por lo que los asignamos a una entrega.

**Liberado:** Se trata de la entrega de los archivos al repositorio en el que se van almacenando las versiones.

**GITHUB**

Github es una plataforma creada para facilitar el desarrollo colaborativo de software, nos permite alojar proyectos como repositorios en la web gratuitamente, por lo general de forma pública, aunque podemos alojar los proyectos de modo privado, si pagamos una pequeña suscripción mensual.

Cuenta con una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero.

Github tiene dos modalidades: una gratuita y otra de pago. En la versión gratuita, se podrán crear “n” repositorios de acceso público; pero para el uso de repositorios privados tendrá un coste.

### 1.4.2 Entorno

El ambiente de trabajo estará compuesto por:

**Equipo de desarrollo:** Las personas involucradas en el proyecto, tienen acceso para poder modificar los documentos del repositorio.

**Administrador:** Se encarga de verificar los cambios de los documentos, y revisar que se trabaja en el branch establecido. Dará los permisos necesarios al equipo para realizar el desarrollo.

**Repositorio Remoto de producción y pruebas:** Github.

### 1.4.3 Infraestructura

Se manejará 2 tipos de ramas o branch, que servira para controlar mejor los commit, se desarrolla en el branch development que corresponderá 1 branch para cada miembro del equipo y la rama maestra (branch master) la cual alojará las versiones cuyos cambios hayan sido aprobados y estén listos.

**Branch master:** Esta rama será la principal, donde se pondrá los cambios aprobados por el administrador.

**Branch Development:** Esta rama será para los desarrolladores, donde podrán hacer sus cambios previa aprobación, sin alterar la rama principal.