**Plan de Gestión de Configuración de Software**

**El documento describe el Plan de gestión de la configuración que se seguirá en el desarrollo del Sistema de Administración de la Cooperativa Puntos Verdes**

**Índice**

1. **Introducción**

La gestión de la configuración es uno de los componentes de éxito de los proyectos TIC. Sin una gestión de la configuración efectiva, la integridad de los ítems de configuración del proyecto y la capacidad de reportar el estado y configuración de aquellos ítems se pone en peligro.

El plan de gestión de la configuración permite establecer un método consistente para identificar y controlar formalmente los ítems de configuración del proyecto.

Los ítems de configuración del proyecto incluyen elementos de hardware y software, así como también información de gestión del proyecto como planes.

La gestión de la configuración es una función integral de la provisión de proyectos TIC porque facilita la protección de los ítems de configuración y comunica los cambios que se han hecho sobre ellos.

Una gestión de la configuración, planificada y ejecutada de manera efectiva, contribuye a la producción de productos TIC de alta calidad evitando el retrabajo. Esto aumenta el valor de los activos informáticos y ahorra costes, contribuyendo a la entrega de proyectos que satisfacen los costes, calendarios, calidad y requerimientos establecidos.

* 1. **Propósito**

El objetivo de este Plan de Gestión de la Configuración (CM), es proporcionar una visión general de la organización, actividades, tareas en general, y los objetivos de Gestión de la Configuración.

Un plan de gestión de la configuración sirve como una herramienta medular de planificación que describe los esfuerzos de planificación para implementar y ejecutar la gestión de la configuración a lo largo del ciclo de vida del proyecto. Provee visibilidad y control del producto referente a su desempeño, funcionalidad y atributos físicos.

* 1. **Aplicabilidad**

El plan de gestión de la configuración debe ser desarrollado para cualquier proyecto clasificado mediano y pequeño.

* 1. **Gobierno**

El Líder de proyecto es responsable de asegurar que el plan de gestión de la configuración sea desarrollado en conjunto con el plan de gestión del proyecto. El Lider de proyecto asegurará su integración en la planificación general.

* 1. **Alcance**

El plan de gestión de la configuración debe involucrar a todas las fases del ciclo de vida del software. El documento permitirá mostrar los estándares de etiquetación de los productos de trabajo.

Así mismo esclarecerá el tipo de nomenclatura utilizada para el control de las versiones de los documentos que se encuentran dentro de los elementos de la gestión.

Por otra parte cualquier stakeholder podrá presentar cualquiera de los siguientes tipos de peticiones de cambio sobre el sistema, para el control de cambios:

* Petición de cambios en los requerimientos (adiciones, supresiones, modificaciones, aplazamientos) en el software actualmente en desarrollo
* Informes de los problemas en la producción corriente o sistemas de pruebas beta
* Petición de mejoras en los sistemas actuales de producción
* Petición de nuevos proyectos de desarrollo.

Este proceso de control de cambio se aplica a los productos de línea base creados o gestionados por los miembros del sistema, incluyendo:

* El software que se ha lanzado a la producción o se encuentra en versión beta
* Requisitos de las especificaciones del sistema
* Grupo de procedimientos y procesos
* Usuarios y documentación técnica

Las siguientes clases de productos de trabajo están exentos de este proceso de control de cambios:

* Los productos de trabajo que están todavía en desarrollo, a excepción de cambios en los requerimientos solicitados en nuevos proyectos
  1. **Definiciones**
* Línea Base. Un documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.
* Petición de Cambio. Una solicitud que alguien ha presentado al sistema de control de cambio que describe un problema de software, una mejora solicitada, una propuesta de cambio en los requisitos de un producto en fase de desarrollo, o un nuevo proyecto que se propone.
* Stakeholder. Persona que directa o indirectamente se ve afectada por el sistema y que puede afectar el proyecto.
* Configuration Item. Los elementos que son puestos bajo el control de gestión de la configuración.
* Control de cambio. Proceso renovable de controlar todos los cambios.

1. **Gestión de la SCM**
   1. **Organización**
   2. **Roles y responsabilidades**

El equipo de la Gestión de Configuración está conformado por los siguientes roles:

|  |  |
| --- | --- |
| **CM Rol** | **Responsabilidades** |
| Jefe del proyecto | * Definir los componentes de desarrollo. * Definir el control de accesos. * Definir políticas generales. * Definir la consecución de hitos. * Asignar actividades. |
| Administrador de la Configuración | * Crear los repositorios de administración de configuraciones. * Configurar entornos para los proyectos * Implementar las políticas definidas por el Jefe del proyecto . |
| Administrador de Releases | * Implementar los pasos de los elementos al entorno de producción a partir de las líneas base creadas por el Sistema de Administración de Configuraciones |
| Integrador del proyecto (Fábrica de Desarrollo) | * Identificar posibles construcciones en el entorno de desarrollo. * Entregar los requerimientos implementados de manera formal para su integración. * Sincronizar cuando sea necesario los cambios realizados por la fábrica con el punto de entrada al Sistema de Administración de Configuraciones. |
| Integrador del proyecto | * Identificar posibles construcciones para los entornos de pruebas y producción. * Planificar las liberaciones de las línea base del producto * Establecer liberaciones * Seleccionar la ubicación para artefactos liberados |

* 1. **Políticas, directrices y procedimientos**
* Cada usuario tendrá derecho a solicitar un repositorio Git para su uso personal, el cual estará restringido por una quota y podrá distribuirla de la manera que más le acomode, pudiendo dividirla entre los repositorios que estime conveniente.
* Cada repositorio Git asignado a grupo de trabajo para los ramos impartidos por el departamento tendrá reglas diferentes a las de los personales y serán controlados por el Administrador de la configuración.
* Cada repositorio Git que sea asignado para un ramo solo durara desde el inicio del ramo hasta el final del ramo. Luego se dará un semestre para que lo respalden, luego de eso será eliminado.
  1. **Herramientas, entorno e infraestructura**

Como herramienta se utilizara el sistema de control de versiones Git, para el entorno de desarrollo se utilizara Github.

**Herramientas:** Git es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

Entre las características más relevantes se encuentran:

* Fuerte apoyo al desarrollo no-lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no-lineal. Una presunción fundamental en Git es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.
* Gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
* Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH. Git también puede emular servidores CVS.
* Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
* Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio.
* Resulta algo más caro trabajar con ficheros concretos frente a proyectos, eso diferencia el trabajo frente a CVS, que trabaja con base en cambios de fichero, pero mejora el trabajo con afectaciones de código que concurren en operaciones similares en varios archivos.
* Los renombrados se trabajan basándose en similitudes entre ficheros, aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de sistema de ficheros, lo que evita posibles, y posiblemente desastrosas, coincidencias de ficheros diferentes en un único nombre.

**Entorno:** GitHub es un software web para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub

Entre sus principales características se puede mencionar:

* Wiki para cada proyecto.
* Página web para cada proyecto.
* Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto.
* Funcionalidades como si se tratase de una red social

**Infraestructura:**



* 1. Calendario