**Plan de Gestión de la Configuración**

**Tío Rico**

**Versión 1.1**

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 01/05/2019 | 1.0 | Creación del documento  Completar la introducción y la gestión de la SCM. | Todo el equipo |
| 02/05/2019 | 1.1 | Actualización de Herramientas, Entorno e Infraestructura | Robert Gutierrez |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[1.Introducción 3](#_Toc7654305)

[Problemática de la empresa: 3](#_Toc7654306)

[Propósito del plan: 3](#_Toc7654307)

[Finalidad del plan: 3](#_Toc7654308)

[2. Gestión de la SCM 4](#_Toc7654309)

[Roles, Responsabilidades: 4](#_Toc7654310)

[Políticas, Directrices o Procedimientos: 5](#_Toc7654311)

[Herramientas, Entorno e Infraestructura: 5](#_Toc7654312)

[Calendario: 5](#_Toc7654313)

# 

# **1.Introducción**

## **Problemática de la empresa:**

Una de las problemáticas más comunes que tenemos en nuestra consultora se generan cuando desarrollamos un proyecto de software, pues lo miembros del equipo de desarrollo hacen uso de diferentes repositorios, esto genera que el jefe de proyecto no haga un control eficiente de las versiones de los elementos, además de generar retrasos en las entregas, errores que no se tenían planificados.

Estos problemas se generan por una falta de comunicación entre los miembros del equipo y el jefe de proyecto, así como también la falta de capacitación en el uso de repositorios como github, los miembros del equipo no saben lo que deben hacer cada vez que quieren cambiar algo es por eso que cada uno trabaja por su propia cuenta.

Y en los casos que se trabaja en un mismo repositorio, se nos presenta el problema de combinar los documentos en Word. En ocasiones ocurren conflictos que dificultan el avance del proyecto.

## **Propósito del plan:**

Se desarrolla este plan con el propósito de establecer pautas para un mejor desempeño de los equipos de trabajo.

## **Finalidad del plan:**

El objetivo de este plan es poder llevar un control de los proyectos, recursos y los tiempos para el desarrollo.

Y se logrará a través de este documento, donde se hará mantenimiento de todos los proyectos.

# 2. Gestión de la SCM

## Roles, Responsabilidades:

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Rol** | **Responsabilidad** |
| Planeamiento de la gestión de la configuración | Project Manager | Supervisa el funcionamiento de la gestión de la configuración. |
| Identificación de la configuración | Gestor de configuración | Se responsabiliza principalmente de la definición y la calidad del proceso de gestión de la configuración.  Gestionar la planificación, identificación, control, seguimiento y auditoría de todos los elementos de configuración en la base de datos de configuración.  Desarrollar el plan de gestión de configuración.  Promover el uso efectivo de la base de datos de configuración dentro de la organización.  Monitorear y reportar los cambios no autorizados sobre los elementos de configuración. |
| Control de la configuración | Bibliotecario de configuración | Custodia la información de los artículos de configuración.  Es responsable de la biblioteca del software, repositorio oficial de las líneas base del proyecto en curso.  Controla el ingreso y el acceso a las líneas base, garantizando el uso de los procedimientos formales definidos en el PGC.  Proveer a los desarrolladores las copias de las líneas base requeridos para sus diferentes tareas.  Registrar y mantener copias de las antiguas versiones. |
| Auditorías de la configuración | Auditor de configuración | Se responsabiliza de la ejecución de auditorías de configuración.  Comprueba que efectivamente el producto que se está construyendo es lo que pretende ser.  Verificar que la configuración actual del software corresponda con lo que era en fases anteriores.  Validar que la configuración actual del software satisface la función que se esperaba del producto en cada hito del proceso de desarrollo. |

## Políticas, Directrices o Procedimientos:

|  |  |
| --- | --- |
| **Documentos** | **Origen** |
| Documentos de privilegios | Documentos |
| Documentos de Normativas | Documentos |
| Políticas de empresa | Documentos |
| Estándares de la empresa | Documentos |
| Manual de buenas prácticas | Documentos |

**Lineamientos Generales**

1. Los proyectos deberán ser ejecutado en iteraciones incrementales con una demostración del producto al finalizar cada iteración: con esta política, se conocerá el estado del proyecto, evaluando si los requisitos cumplen con las expectativas del cliente, si la calidad es la esperada, o si hay retrasos; agilizando la toma de decisiones correctivas.
2. Los requisitos se desarrollarán priorizados por el valor aportado al cliente: Esta política permitirá que los objetivos más importantes del proyecto sean atendidos.
3. El control y seguimiento de los proyectos se basará en los requisitos completados en cada iteración. Se entiende como un requisito, los entregables asociados a: análisis, desarrollo, pruebas, documentación, etc. e integrados con los entregables de las iteraciones anteriores. (TioRico/NombreProyecto/Gestión)
4. Cada requisito debe ser independiente del resto de los requisitos, en la medida de lo posible. (TioRico/NombreProyecto/Análisis)
5. Cada requisito debe ser demostrable, permitiendo cómo comprobar con el cliente que el requisito está completado y que se cumplen sus expectativas. (TioRico/NombreProyecto/Análisis)
6. El requisito debe ser de un grado de esfuerzo para ser completado semejante al del resto de requisitos: de manera que la organización y el cliente, pueda realizar una extrapolación del progreso del proyecto. (TioRico/NombreProyecto/Análisis)

**Desarrollo** (TioRico/NombreProyecto/Desarrollo)

1. Los componentes de software deberán ser desarrollados y liberados por partes que serán generados en las iteraciones incrementales y no entregados al final del proyecto.
2. El desarrollo del componente de software que conformarán la solución, deberá ser liberados en varias iteraciones.
3. Cada iteración deberá producir software con calidad de producción, probado, integrado, y documentado (funcional, técnica).
4. Cada iteración deberá cumplir con un subconjunto de requerimientos.
5. Cada iteración deberá contemplar: análisis, diseño, implementación, documentación, etc.

**Pruebas**

1. Cada proyecto debe incorporar las prácticas de TDD (Test Driven Development).
2. Cada uno de los entregables, deberá contener scripts de pruebas unitarias, integrales, funcionales, etc; mediante la utilización de frameworks como junit, dbunit, phpunit, mockObjtects, etc.

**Documentación**

1. La documentación de los proyectos, específicamente:  manual de usuario, manual de operaciones, arquitectura de la solución, especificaciones, etc; deberán ser entregables parciales para cada una de las iteraciones, es decir, la documentación no se liberará al final del proyecto, sino en entregables parciales e incrementales. (TioRico/NombreProyecto/Análisis)

**Control de Calidad**

1. Cada uno de los entregables, serán sometidos a un script de calidad, que ejecutará la organización, y no serán admitidos como productos del proyecto hasta alcanzar un nivel aceptable. (TioRico/NombreProyecto/Gestión)

**Control de Riesgos**

1. Los riesgos serán identificados en la primera iteración, llevándose a cabo también una valoración inicial de la exposición al riesgo y planes de contingencia. En cada iteración se revisará y actualizará el documento “Lista de Riesgos”, añadiendo además la lista de riesgos más importantes actualizada por cada iteración. (TioRico/NombreProyecto/Gestión)

**Control de Artefactos**

1. Cada uno de los artefactos del proyecto, deberán ser mantenidos bajo un sistema de control de versiones.
2. La organización dispondrá de un sistema de control de versiones, que deberá ser actualizado por el cliente de forma remota.

## Herramientas, Entorno e Infraestructura:

Durante el proceso de gestión de configuración se utilizará la herramienta  Git para el control de versiones del producto. Cuando algún miembro haga una modificación en el proyecto,  deberá acceder al servidor donde está alojada el repositorio de la empresa para almacenar la parte  modificada en él, teniendo el resto del equipo de desarrollo la última versión actualizada en dicho servidor. Esta gestión de acceso al servidor para la actualización se hará mediante la plataforma Github tanto para los documentos y código fuente.

El control de cambios, así como la gestión de defectos, se llevará a cabo mediante la funcionalidad de incidentes (issues) de la plataforma GitHub, permitiendo al equipo de desarrollo ir detallando los defectos encontrados para que la persona responsable pueda corregirlos.

Todos los miembros del equipo de trabajo tendrán acceso a estas herramientas.

* Git 2.21 como herramienta de gestión de configuración del software.
* Microsoft Office 365 para la documentación en general.

## Calendario:

|  |  |  |
| --- | --- | --- |
| Actividades | Tiempo en Días | Rol |
| Planeamiento de la gestión de la configuración | 7 | Project Manager |
| Identificación de la configuración | 5 | Gestor de configuración |
| Control de la configuración | 23 | Bibliotecario de configuración |
| Auditorías de la configuración | 17 | Auditor de configuración |