**Plan de Gestión de la Configuración**

**Consultora ElmerX**

**Versión 1.1**

**Historial de Revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
| 14/09/2018 | 1.0 | Creación del documento | Steve Sader Matos Manguinuri |
| 14/09/2018 | 1.1 | Introduccion Agregada | Todo el equipo |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**ÍNDICE**

[1. Planificación 3](#_Toc525265521)

[1.1. Introducción 3](#_Toc525265522)

[1.2. Roles , responsabilidades y cantidad 3](#_Toc525265526)

[1.3. Políticas, Directrices y procedimientos 4](#_Toc525265527)

[1.4. Herramientas, entorno e infraestructura 4](#_Toc525265528)

[1.5. Calendario 4](#_Toc525265529)

# 

# Planificación

## Introducción

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.

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

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.

## Roles, responsabilidades y cantidad

|  |  |  |
| --- | --- | --- |
| Roles | Responsabilidades | Cantidad |
| Gestor de la configuración | Se encargara de gestionar la infraestructura global de la gestión de la configuración y el entorno del equipo de desarrollo de los productos | 2 |
| Jefe de proyecto | Se encargara de gestionar y velar por el correcto desarrollo del proyecto en su totalidad. | 4 |
| Gerente funcional | Se encargara de aprobar y vigilar la utilización de los recursos para los proyecto sin afectar notoriamente el capital o recursos internos de la empresa. | 1 |
| Director de programa | Se encargara de gestionar programas ( conjunto de proyectos relacionados) de forma óptima sin que los proyectos sean afectados entre sí. | 1 |
| Miembros del equipo | Personas que formaran parte del equipo operativo de los proyectos. | 20 |

## Políticas, Directrices y procedimientos

|  |  |
| --- | --- |
| Documento | Origen |
| Documento de privilegios | Documentos |
| Manual de procesos | Documentos |
| Políticas de la empresa | Documentos |
| Estándares de la empresa | Documentos |
| Manual de buenas practicas | Documentos |

**Lineamientos Generales**

1. El proyecto deberá 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. El proyecto se ejecutará en iteraciones incrementales con una duración fija de 3 semanas.
3. Los requisitos se desarrollarán priorizados por el valor aportado al cliente: Esta política permitirá que los objetivos más importantes del proyecto sean atendidos.
4. El control y seguimiento del proyecto 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.
5. Cada requisito debe ser independiente del resto de los requisitos, en la medida de lo posible.
6. Cada requisito debe ser demostrable, permitiendo cómo comprobar con el cliente que el requisito está completado y que se cumplen sus expectativas.
7. 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, puedan realizar una extrapolación del progreso del proyecto.

**Desarrollo**

1. Los componente de software, deberán ser desarrollados y liberados por partes, y no entregados al final del proyecto.
2. El desarrollo de los componente de software que conformaran la solución, deberán 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 practicas 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, mockObjtects, etc.

**Documentación**

1. La documentación del 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 liberara al final del proyecto, sino en entregables parciales.

**Control de Calidad**

1. Cada uno de los entregables, serán sometidos a un script de calidad, que ejecutara la organización,  y no serán admitidos como productos del proyecto hasta alcanzar un nivel aceptable.

**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.

**Control de Artefactos**

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

## Herramientas, entorno e infraestructura

* Git
* Spring Framework
* GlassFish
* Android Studio
* Postgresql
* Heroku
* Amazon Web Services (AWS)

## Calendario

|  |  |  |  |
| --- | --- | --- | --- |
| ACTIVIDAD | | TIEMPO (días) | ROL |
|  | |  |  |
| 1 | **Planificación de la SCM** |  |  |
|  | Identificar la problemática de la empresa | 1 | Gestor |
|  | Definir el propósito y finalidad del plan | 1 | Gestor |
|  | Identificar políticas y procedimientos | 2 | Gestor |
|  | Analizar herramientas | 2 | Gestor |
|  | Definir tareas | 5 | Project Manager |
|  | Definir equipos y roles | 3 | Project Manager |
|  | Capacitar al equipo | 7 | Miembros del equipo |
|  | |  |  |
| 2 | **Identificación de la SCM** |  |  |
|  | Definir nomenclatura de elementos | 2 | Project Manager |
|  | Definir la estructura del repositorio | 2 | Project Manager |
|  | Plantear privilegios y seguridad | 2 | Auditor |
|  | Definir elementos | 4 | Gestor |
|  | Clasificar elementos | 2 | Gestor |