

Proyecto: Calculadora

Plan de Administración de Configuración

**Autores: Carolina Gatica Rodriguez
Franz Yépez Hinostroza**

1. INTRODUCCION

1.1 Propósito y alcance

Este documento trata sobre el plan de Administración de la Configuración (CM) para el “Proyecto Calculadora”. La intención del plan de CM es el control de la configuración de los requerimientos, documentos, software y herramientas usadas para este proyecto. Las herramientas usadas son de código libre, pertenecientes a distintos entes.

1.2 Propósito de las prácticas de SCM

- Asegurar consistencia en las actividades prácticas de SCM
- Definir los organismos competentes para apoyar las prácticas de SCM
- Mantener la integridad del producto durante su ciclo de vida
- Informar a los grupos e individuos involucrados sobre el estado del proyecto
- Crear un historial de los estados actuales y anteriores de los productos en desarrollo
- Mejorar procesos

1.3 Referencias / Siglas / Glosario

CCB : Change Control Board
CM: Configuration Management
PCM: Project Configuration Management
SCM: Software Configuration Management
Tag: Etiqueta
CI: Configuration Item

1.4 Herramientas del CM

- Git: software de control de versiones.
- GitHub: plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones de Git.
<https://github.com/>
- Travis: herramienta para la integración continua
<https://travis-ci.org/>
- Gradle: herramientas para la automatización de pruebas.
<http://gradle.org/>
- Issues: herramienta de control de defectos. Lleva un registro de los defectos o cosas pendientes del proyecto. Propio de GitHub.

2. Puestos de la Administración de la Configuración

las tareas de la administración de la configuración serán coordinadas dentro del proyecto Calculadora por el cargo del Administrador General de la administración de configuración (GPCM) del proyecto, el cual será asignado a una persona. Adicionalmente, un grupo de respaldo será asignado dentro del equipo.

El GPCM será responsable de actividades como hacer seguimiento del main y de las entregas, determinar las ramas a ser creadas, que actividades de desarrollo pertenecen a la rama, etc.

También los equipos de desarrollo tendrán su propio Team Project Configuration Manager, el cual responderá tanto al CM como al GPCM, colaborando dentro del alcance de sus responsabilidades.

Las actividades, procesos, procedimientos y políticas de la CM deberán ser seguidas por todos los miembros del equipo. Es responsabilidad de cada persona seguir y aplicar el proceso correcto de CM de acuerdo a su/s cargo/s asignado/s.

2.1 Administradores de la Configuración del Proyecto

Puesto	Principal	Vice
Global PCM	Carolina Gatica Rodríguez	Franz Yépez Hinostroza

2.2 Responsabilidades en la Administración de la Configuración

- GPCM:
 - Entera responsabilidad de toda la configuración.
 - Responsabilidad sobre la creación de todas las ramas y administración de sus políticas.
 - Responsabilidad por la aplicación de etiquetas en el main y en las entregas.
 - Asegurar la integridad del producto y el seguimiento de los items de configuración de todo el proyecto.
 - Coordinar las actividades de CM dentro del proyecto.
 - Asegurar la correcta ejecución del esquema de CM.
 - Asistir en las actividades de mergeo del main y entregas.
 - Construcción de las actividades del main y los niveles de entrega.
 - Participar en las auditorías.
 - Analizar todas las conclusiones referidas al CM.
- TPCM:
 - Asistir en la creación de etiquetas y ramas.
 - Construir ramas de actividades específicas del equipo.
 - Asegurar la integridad del producto y el seguimiento de los CI propios del equipo.
 - Participar en auditorías.
 - Analizar todas las conclusiones referidas al CM.
- Team: Ayudar a resolver conflictos durante las actividades de mergeo.
 - Asegurar que se cumplan los criterios de calidad de las entregas a la rama principal.
 - Hacer seguimiento de todos los procesos, políticas y prácticas para cada rol designado.

3. Administración del cambio

Change Control Board

3.1 Propósito

La CCB es una junta que asegura que todo cambio es apropiadamente considerado por todas las partes y autorizado antes de su implementación. La administración del cambio tiene la intención de asegurar que la evolución del sistema sea un proceso gestionado en el que se da prioridad a los cambios más urgentes y rentables.

La CCB es responsable de aprobar, monitorear y controlar los pedidos de cambio para establecer líneas bases de los ítems de configuración.

En este proyecto en particular, las peticiones de cambio son originadas por el cliente y todas ellas fueron consideradas válidas. Ante estas solicitudes, la CCB analizó la factibilidad de la implementación de estos cambios, los cuales consistían en agregar nuevas funciones al proyecto.

Debido a que trabajamos para un solo cliente, es de vital importancia tener en cuenta sus peticiones de cambio en el momento de la toma de decisiones de la CCB. Los cambios propuestos sólo afectan a un solo individuo, quien los propuso en primer lugar.

El costo a tener en cuenta por la CCB es en relación a las horas de trabajo dedicadas a la programación de las nuevas funcionalidades requeridas, y en este caso se estima relativamente bajo.

3.2 Integrantes de la CCB

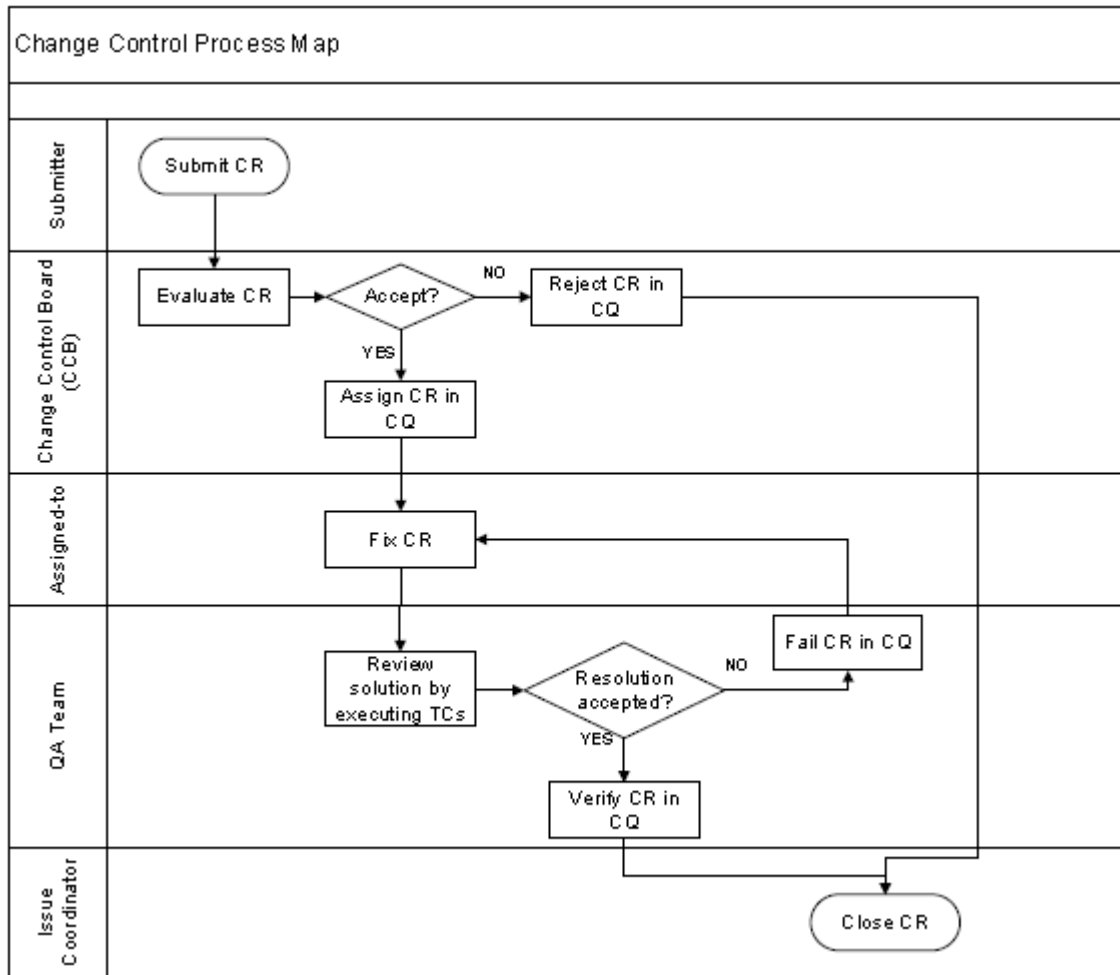
Administrador de Ingeniería- CCB Chair
Administrador de Entregas – Coordinador de Fallas
Director de Ingeniería
GPCM

La forma de contacto de los integrantes del equipo es a través de mail, celular y redes sociales en general.

3.3 Reuniones

La CCB se reúne cada vez que surja una nueva petición de cambio o problemas a solucionar en el software, por lo cual las reuniones no tienen una periodicidad definida. Es de vital importancia que se presenten el GPCM y Vice GPCM para cualquier toma de decisión.

3.4 Esquema del proceso de control de cambio



4. Identificación de Configuración

El repositorio de github <https://github.com/alexfyh/Tp1--SOFT-.git> será usado por todos los miembros del equipo de desarrollo para guardar y compartir recursos de las diferentes versiones de proyecto.

Por cada realización hecha en el repositorio, se actualizará con los datos pertinentes en el archivo ReleaseNotes.txt.

5. Definición de Ramas:

- Rama de integración: el Master es definido como la rama principal de integración donde todas las nuevas funcionalidades serán añadidas.
- Ramas de desarrollo: se usarán para desarrollar individualmente una nueva función del software sin que altere la estabilidad de la versión que se encuentre en la rama principal. Una vez que el desarrollo de la nueva función se encuentre finalizado, se procede al mergeo con la rama principal.
El formato del nombre es: rama<nueva_funcionalidad>.

6. Definición de etiquetas:

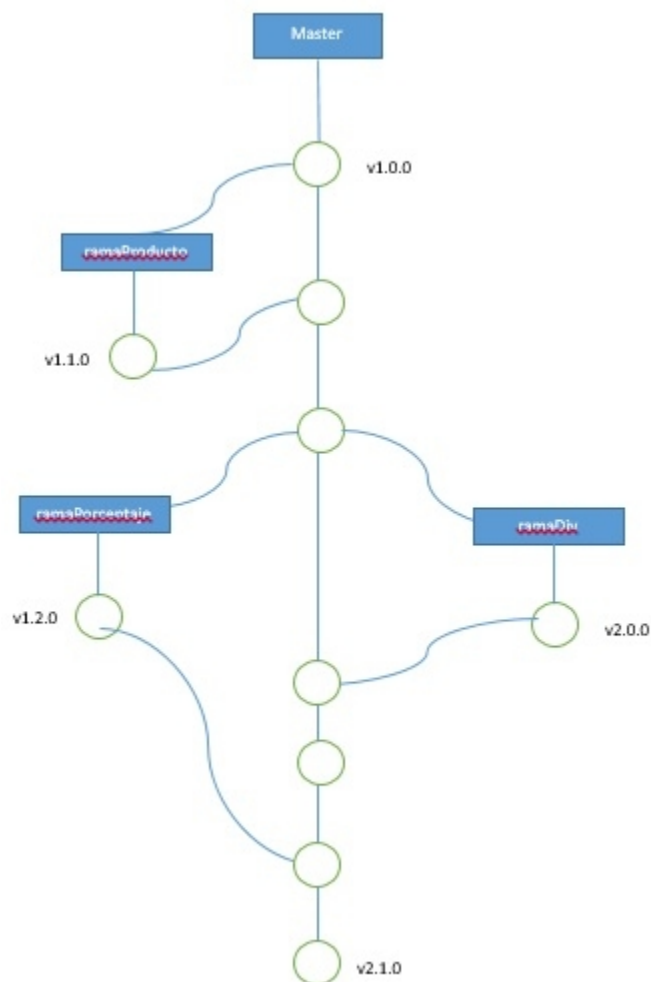
- Mergeo de etiquetas: se sugiere cada vez que un mergeo es realizado en la versión del código. El formato es merge-from-<source>-<ddmmaaaa>.
- Etiquetas de versión: se usa cada vez que un release es hecho en el proyecto. El formato es por ejemplo v.1.0.0.
De acuerdo al grado de modificación del código, corresponderá el cambio del número de la versión, correspondiendo el primero a cambios de versiones muy significativos, el segundo a variaciones consideradas menos importantes, y el tercero a arreglo de bugs.

Archivos auxiliares de CM

- MergesNotes.txt: En ella se detallará de manera concisa las acciones de mergeo producidas durante el desarrollo del código. En ella se especificarán las ramas involucradas, la fecha, el desarrollador que la llevó a cabo, así como si el mergeo dio origen a la conclusión de alguna de las ramas mencionadas.
Formato:
DD/MM/AAAA (Nombre del desarrollador) Merge from (branch) to master
Comentario
- ReleaseNotes.txt: En ella se detallará el historial de release del proyecto.
Formato:
Fecha: DD/MM/AAAA
Autor:
Titulo:
Comentarios:

7. Estrategias de Mergeo

Se realizará el mergeo cada vez que haya se agregue una funcionalidad que será reutilizada. Se debe mergear sólo cuando la nueva funcionalidad de la rama a mergear se encuentre completa.



8. Nivel de calidad y Criterios de realización

Se seguirá una rama principal, llamada master, cuyo nivel de calidad corresponderá al necesario para realizar un release en la versión correspondiente del proyecto. Ante una solicitud de cambio aprobada del cliente, se procederá a crear una nueva rama de desarrollo en la que el nivel de calidad puede variar antes de su finalización. Antes de ser mergeada a la rama master, es necesario que el código cumpla con los criterios de realización propios de la rama master, lo que implica que el nivel de calidad de la rama principal siempre

se mantenga. Si luego del mergeo los Unit Test advirtieron un resultado no esperado , se procederá a corregir el problema. Si no hubiera una solución, la rama principal volverá a una versión anterior al mergeo.

9. Build Management

Builds de Integración continua: Son ejecutados en aquellas ramas que demandan que el código no se rompa durante su desarrollo. Son útiles para identificar lo más temprano posible errores de código antes de su construcción formal . Se basan en el testo automático. Por cada commit se corren test que generan reportes sobre el estado actual del proyecto. En este caso fue utilizado Travis-CI en sincronización con el repositorio GitHub donde se encuentre el proyecto Calculadora. Los test son ejecutados por Gradle.