

# SCM

## Libro: “Elements of Software Configuration Management”

### **SCM en contexto**

En orden para entender apropiadamente el rol que el SCM juega en el proceso de desarrollo de un software, debemos primero entender qué es el objetivo de un proceso. El objetivo de un desarrollador de software es, o debería ser, la construcción de un producto que se acerque a lo que realmente necesita un grupo de personas para quienes el software es desarrollado. Llamamos a este objetivo el logro de la integridad del producto. Esta integridad tiene un grupo de atributos intrínsecos que caracterizan al producto:

- que cumple con las necesidades funcionales del usuario
- que puede ser trazado fácil y completamente a través de su ciclo de vida
- que conoce los criterios específicos de performance
- cuyas expectativas de costo son conocidas
- cuyas expectativas para entregar son conocidas

Hemos aprendido que igualar los términos de “software” y “programa de computadora” restringe inapropiadamente nuestra visión de software. Entonces para nuestra definición, software no es simplemente un conjunto de programas de computadora, sin que incluye la documentación requerida para definir, desarrollar y mantener estos programas.

Los desarrolladores son, en realidad, un conjunto complejo de entidades organizacionales interactuantes. Muchos desarrolladores se estructuran a sí mismos dentro de tres grupos de disciplinas básicas que incluyen:

- Project management: dan soporte general a las necesidades de mantenimiento para ver que esta pasando en un proyecto para garantizar que la organización principal desarrolle sistemáticamente productos con integridad. Al mismo tiempo, estas disciplinas miran dentro del proyecto para dar soporte a la asignación, reparto y control de todos los recursos del proyecto.
- Development: representa las disciplinas tradicionalmente aplicadas a un proyecto de software. Incluye: análisis, diseño, ingeniería, producción, test, instalación, documentación, entrenamiento y mantenimiento. Toma un bien estructurado enfoque técnico riguroso para el desarrollo del sistema, a lo largo con una buena mezcla de disciplinas de desarrollo para buscar la integridad del producto, especialmente del software.
- Product assurance: las disciplinas que son usadas en el mantenimiento de un proyecto para ganar visibilidad en el proceso de desarrollo son:
  - Configuration Management (CM): es la disciplina para identificar la configuración de un sistema en puntos discretos a tiempo con el propósito de control sistemáticamente los cambios de configuración y manteniendo la integridad y trazabilidad de la configuración a través del ciclo de vida del sistema.
  - Quality assurance (QA): es la consistencia de procedimientos, técnicas y herramientas aplicadas por profesionales para asegurar que el producto conoce o excede estándares preespecificados durante el ciclo de desarrollo de un producto. Implica asegurar que un producto conoce o excede un mínimo nivel aceptable de excelencia industrial y/o comercial.
  - Validation and Verification (VyV): se ha creado expresamente para hacer frene al software y su desarrollo. Trata con el tema de cómo el software cumple los requerimientos funcionales y de rendimiento y garantiza que son declarados e interpretados correctamente.
    - Verificación: asegura que el producto conoce los objetivos prescritos como fueron definidos a través de la documentación de la línea base.
    - Validación: asegura que el producto haga el trabajo correcto. Es invocado para asegurar que el usuario final tenga el producto correcto.
  - Test and Evaluación (TyE): disciplina impuesta desde fuera de la organización del desarrollo del producto para independientemente evaluar si el producto cumple con los objetivos. Implica evaluar el rendimiento del producto en un ambiente real o cercano al real. La filosofía dice que los desarrolladores no pueden ser objetivos en el momento de evaluar o testear lo que ellos produjeron.

El mantenimiento debe preocuparse en minimizar la aplicación excesiva o redundante de recursos para direccionar el solapamiento entre estas disciplinas.

## Los elementos del SCM

Los elementos del SCM fueron elegidos para ser los mismos que para el CM de hardware. Como el de hardware, los cuatro componentes del SCM son:

### ➤ Identificación:

Un mantenimiento efectivo del desarrollo de un sistema requiere una definición cuidadosa de los componentes de la línea base; cambios en estos componentes también necesitan ser definidos desde estos cambios, junto con las líneas base, especificar la evolución del sistema.

Una línea base puede ser caracterizada por dos etiquetas. Una etiqueta identifica la línea base en sí y la otra identifica una actualización de una línea base particular. Cuando los cambios son hechos en la línea base establecida más reciente, luego, desde el punto de vista del mánager de la configuración de software, esta línea base y estos cambios encarnan al actual software en su más reciente estado.

La entidad más elemental del mecanismo de etiquetado de la identificación es el “ítem de configuración (SCI)”. Al realizar la función de identificación, el mánager hace, en efecto, tomar fotos de los SCI. La colección de los estados del ciclo de vida parece una colección de secuencias escalonadas y solapadas de fotos de los árboles de los SCI. La identificación de líneas base y actualizaciones provee un camino de documentación explícita que une todos los estados del ciclo de vida de un software.

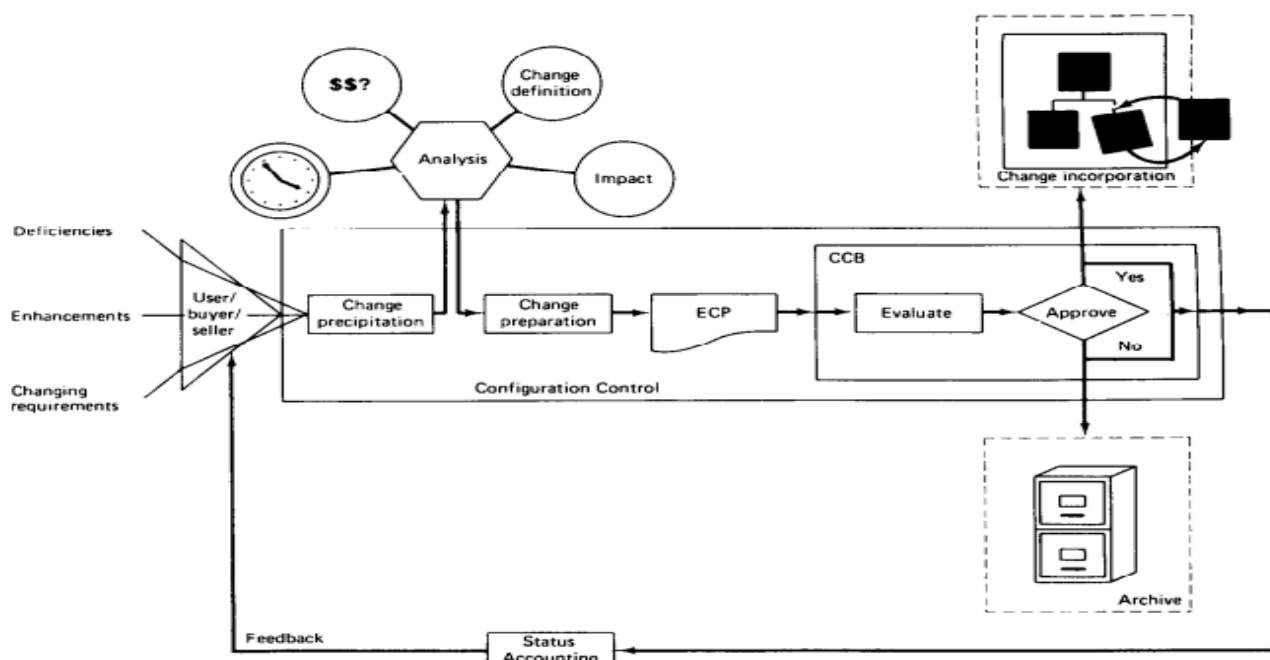
### ➤ Control:

Uno de los roles de este elemento es proveer de un mecanismo administrativo para precipitar, preparar, evaluar y aprobar o desaprobar todas las propuestas de cambio que pasan por el ciclo de vida del sistema.

En líneas base tempranas y sus respectivas actualizaciones, sus SCI son documentos de especificación, luego en líneas base más adelantadas, cada SCI puede manifestarse en alguna o todas las variadas representaciones del software. El control se enfoca en manejar los cambios de todas las representaciones de los SCI. Este proceso involucra tres ingredientes básicos:

- 1) Documentación para formalmente precipitar y definir una propuesta de cambio al sistema de software.
- 2) Un cuerpo organizacional para formalmente evaluar y aprobar o desaprobar una propuesta de cambio al sistema de software (el Comité de Control de Configuración - CCC).
- 3) Procedimientos para controlar los cambios al sistema de software.

La ECP (Engineering Change Proposal – Propuesta del Cambio de Ingeniería), el documento de control mayor, contiene información como la descripción del cambio propuesto, identificación de la organización que origina la propuesta, razonamiento del cambio, identificación de las líneas base afectadas y sus SCI, y una especificación del costo y el impacto en el esquema. Este documento es revisado por el CCC.



La incorporación del cambio no es una función del SCM, pero el monitoreo del resultado del proceso de implementación del cambio si lo es.

Muchas herramientas automatizadas soportan el proceso de control. Una de las mayores ayudas una vez que la etapa de código fue alcanzada, y que son referidas genéricamente son las librerías de soporte del programa (PSL). Una PSL puede proveer un legible y centralizado repositorio para versiones autorizadas de cada componente del sistema.

Un PSL puede soportar tres actividades principales: desarrollo de código, mantenimiento del software, y el control de configuración. Soportar el proceso de desarrollo incluye el soporte de diseño, codificación, testing, documentación y el mantenimiento del programa a lo largo con el esquema y subesquema de la base de datos asociada.

Un PSL tiene cuatro componentes: librerías internas legibles para las máquinas, librerías externas en copias externas, procedimientos computacionales y procedimientos de oficina. Los componentes de un sistema PSL están interbloqueadas para establecer una correspondencia exacta entre las unidades de código internas y las versiones externas de los sistemas de desarrollo. Esta continua correspondencia es una característica que garantiza una visibilidad continua y la identificación del desarrollo del sistema.

#### ➤ Auditoría

Provee un mecanismo para determinar el grado en el cual el estado actual del software es igual al software pintado en la línea base y en la documentación de los requerimientos. También provee un mecanismo formal para establecer una línea base.

Sirve a dos propósitos:

- Verificación: asegura que cada uno del SCI que se especifican en una línea base (o actualización) este hecho en la siguiente línea base.
- Validación: asegura que la configuración del SCI resuelva el problema correcto.

Esta intencionada en incrementar la visibilidad del software y en establecer una trazabilidad a través del ciclo de vida del producto software. La auditoría cuesta tiempo y dinero.

El resultado de una auditoría puede ser el establecimiento de una línea base, la redirección de las tareas del un proyecto, el ajuste de la aplicación de los recursos del proyecto.

#### ➤ Informes de estado

Es el seguimiento y reporte administrativo de todos los ítems de software formalmente identificados y controlados. Guarda la actividad asociada a las otras tres funciones del SCM y para ello provee el significado de cómo la historia del ciclo de vida del software puede ser monitoreada. Todas estas cosas pueden ser grabadas por un PSL automatizado.

Es una función que incrementa en complejidad a medida que el ciclo de vida progresa porque múltiples representaciones del software emergen con líneas base subsecuentes.

## Libro: “Little Book of Configuration Management”

### El Proceso de CM

El propósito fundamental del Mantenimiento de Configuración (CM) es establecer y mantener la integridad y el control de los productos de software a través del ciclo de vida de un proyecto. CM incluye identificar la configuración del software en ciertos puntos de tiempo, sistemáticamente controlar cambios en la configuración, y mantener la integridad y trazabilidad de la configuración a través del ciclo de vida del proyecto.

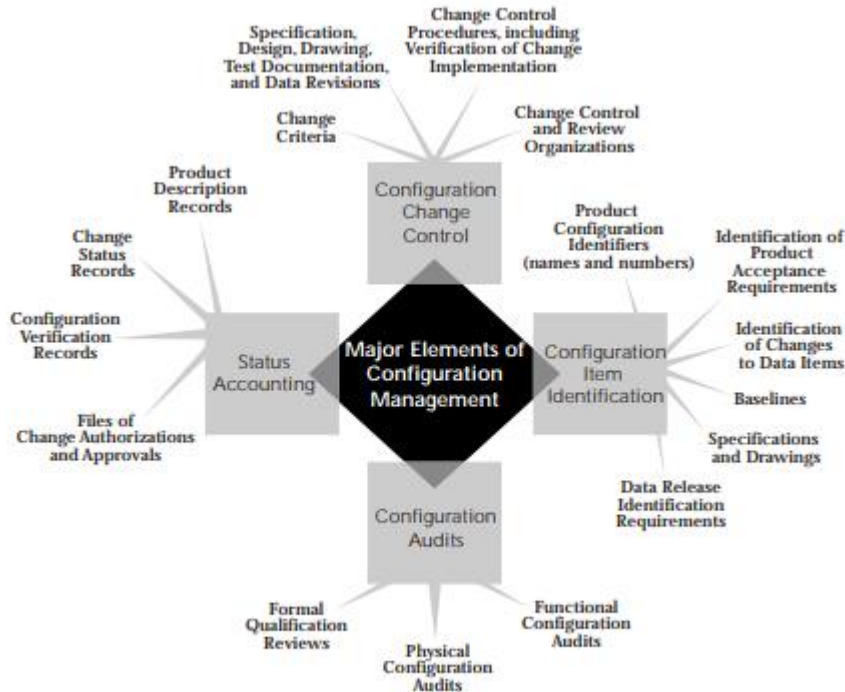


Figure 1. Major Elements of Configuration Management

### Bases del CM

- Identificación de la configuración: habilidad para identificar qué información ha sido aprobada para uso concurrente en el proyecto, quién posee la información, cómo la información fue aprobada por el control de CM y las últimas releases aprobadas.
- Control: designan el nivel de control por el cual cada producto de trabajo debe pasar; identificar las personas y grupos con autoridad para autorizar cambios y para hacer cambios en cada nivel; y los pasos a ser seguidos para obtener esa autorización requerida para los cambios, para procesar las peticiones de cambios, para seguir los cambios, para distribuirlos y mantener las versiones pasadas.
- Informes de estado: la información del registro proporciona un accesible y actual guardado de estado de cada pieza de información controlada que está planeada en usarse.
- Reseñas y auditorías: una frecuente evaluación del contenido, la integridad de la línea base y la integración de los releases de todos los productos para asegurar que se ajustan a la documentación de configuración.

### Actividades primarias del CM

CM aplica dirección y vigilancia técnica y administrativa a:

- Identificar y documentar las características funcionales y físicas de un sistema, software, hardware o un componente operacional para que estas relaciones puedan ser manejadas, mantenidas, controladas y aseguradas.
- Control de cambios a esas características
- Registro y reporte del estado de los cambios propuestos, y el estado de la implementación de los cambios aprobados.

- Diseminar la información de la línea base al personal del proyecto, y establecer y mantener un informe de estado y sistema de reporte que registre la línea base, cambios autorizados en la línea base y verificación si los cambios fueron incorporados dentro de la documentación y/o el producto.
- Reseña y auditoría al proceso de CM para asegurar el proceso y la adecuación del control.
- Establecer una jerarquía de información entregable y no entregable.
- Establecer métricas de CM, e indicadores de tendencia y alerta para soportar la evolución de calidad del producto software.

## Armando el CM

Armar un CM requiere una secuencia planeada y ordenada de actividades y tareas. Las siguientes son elementos clave para establecer un programa de CM exitoso:

- 1) Planear y documentar las requerimientos, actividades y responsabilidades del software de CM.
- 2) Establecer un proceso de desarrollo de software genérico e identificar puntos en los cuales las barreras de calidad asegurarán la integridad del producto.
- 3) Sondear y seleccionar herramientas que puedan, sin una mayor modificación, soportar las necesidades del proceso de software.
- 4) Establecer el esquema de identificación de configuración a ser usada para todos los productos de trabajo del software puestos bajo control de configuración.
- 5) Documentar e implementar un método para recolectar, registrar, procesar y mantener la información necesaria para proveer informes de estado de configuración.
- 6) Proveer un framework para asesorar la integridad del proceso de desarrollo y compartir información, y establecer un significado para integrar, calificar y aceptar diversos componentes del sistema.
- 7) Proveer puntos discretos en los cuales el programa, el presupuesto, la calidad del desarrollo deben ser evaluadas.
- 8) Establecer e implementar los procedimientos de configuración de control, designando el nivel de control por el que cada producto de trabajo debe pasar, los que estén autorizados para autorizar para hacer cambios en cada nivel, para procesar peticiones de cambio, para hacer seguimiento de cambios, distribuir cambios y mantener versiones anteriores.
- 9) Establecer y documentar el esquema de identificación de configuración a ser usado por todos los artefactos puestos bajo control.
- 10) A través de la auditoría de configuración, establecer y documentar los planes y procedimientos para auditorías periódicas y evaluaciones del producto de software. En adición a requerimientos de auditoría específicos, una auditoría de configuración debería ser conducida para determinar:
  - a. Si la calidad del producto conoce los requerimientos funcionales y físicos.
  - b. Si los productos son guardados en una librería controlada correspondiente a los registros del CM
  - c. Si los productos y línea base de los SCI son completas y disponibles con respecto al estado de los productos y cambios aprobados de donde son armados.
- 11) Establecer e implementar procedimientos para el empaquetado, guardado, manejo, respaldo, recuperación, entrega de los productos de software.
- 12) Establecer una supervisión y auditoría. La identificación y recreabilidad de artefactos o productos de trabajo antes de los subcontratistas.

## Mejores prácticas

- *“Hacer que el Mantenimiento de Configuración sea un trabajo de todos”*

Desde la concepción del proyecto, crear un ambiente donde la adherencia al proceso de CM establecido es el único camino aceptable para hacer negocios.

Las aplicaciones de ingeniería de software deben ser desarrolladas y mantenidas en un ambiente que permite la comunicación y asegura un flujo suave y controlado de información; y visión de los cambios de desarrollo que afectan al producto.

Experimentación, retrabajo debido al pobre control de información y cambios incontrolados no pueden ser permitidos en un proyecto para ser exitoso. Las disciplinas de CM son las principales reductoras de riesgo un ambiente de proyecto. CM es vital para el éxito de cualquier esfuerzo de software.

- *“Crear un ambiente y proceso de ingeniería para asegurar el CM”*

Esto es para aplicar los controles correctos a cada fase de desarrollo, para que el control de la línea base sea mantenida y el control de cambio sea hermético.

En los niveles más bajos del proceso de planeamiento de la ingeniería de software, los requerimientos del software CM y los procedimientos específicos de mantenimiento y control de procesos son documentados. El planeamiento y proceso de CM deberían ser definidas durante las primeras fases del esfuerzo propuesto. Esta relación es crítica para el éxito del proyecto de software y la calidad de los productos hechos. Si el software es producido en un ambiente de programa sin claridad, inconsistente, ineficiente, la calidad del software será cuestionable a pesar del rigor del desarrollo del software y la efectividad del proceso de ingeniería del software.