

## UNIDAD N° 3

# Gestión de Configuración del Software

---

# Conceptos Introdutorios de la Gestión de Configuración

## El software

Software es un conjunto de programas y la documentación que lo acompaña. La idea del software como información o conocimiento empaquetado a distintos niveles de abstracción, donde el nivel más bajo es el código.

- Información estructurada con propiedades lógicas y funcionales
- Información creada y mantenida en varias formas y representaciones
- Información confeccionada para ser procesada por computadora en su estado más desarrollado

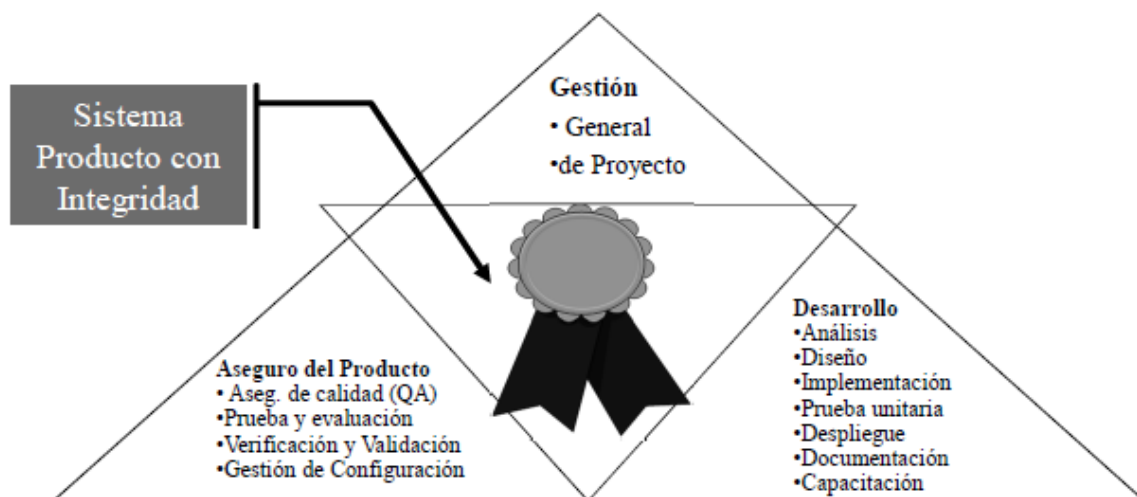
## La Gestión de Configuraciones

La administración o gestión de configuraciones es una disciplina de soporte, forma parte de las disciplinas protectoras y tiene el propósito de mantener la integridad de los productos del proyecto de software a lo largo del ciclo de vida.

Involucra para la configuración:

- Identificarla en un momento dado
- Controlar sistemáticamente sus cambios
- Mantener su integridad y origen

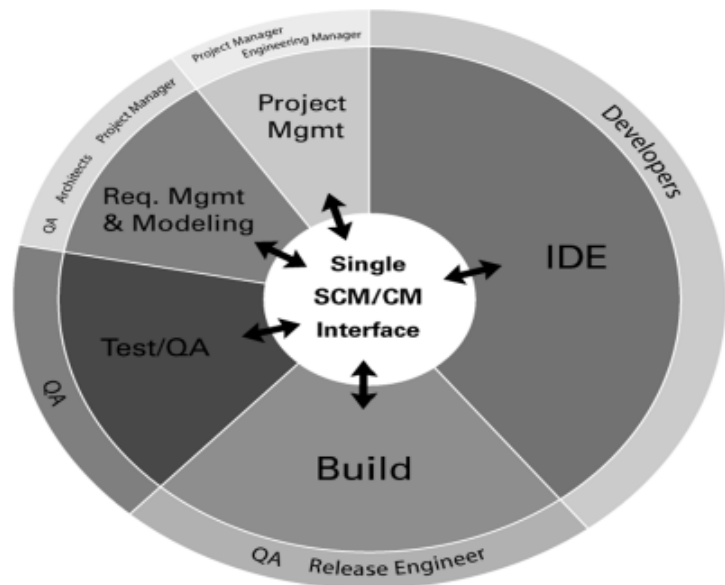
## Triángulo de disciplinas



## SCM como disciplina de gestión

"Es una disciplina que aplica dirección y monitoreo administrativo y técnico a: Identificar las características funcionales y técnicas de los ítems de configuración; Controlar los cambios; Registrar y reportar y Verificar su correspondencia con los requerimientos."

Es transversal a  
**TODO** el proyecto,  
 durante **TODO** su  
 ciclo de vida



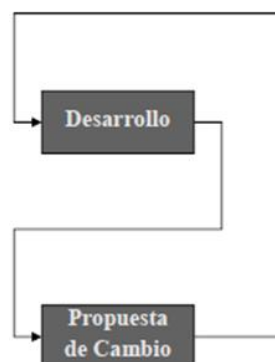
## Integridad de un producto de software

El software es un blanco móvil porque al ser intangible y fácilmente modificable la administración de configuración ayuda a que el producto permanezca inalterable en los términos que fue creado a lo largo del ciclo de vida. Se considera que un producto tiene integridad si:

- **Satisface las expectativas del usuario**  
 Considera los requerimientos funcionales
- **Puede ser fácilmente rastreado durante su ciclo de vida**  
 Puedo saber cómo llego a este punto cada pieza de software, es decir saber a qué requerimiento está asociado y saber dónde se analizó, donde se diseñó, donde se testeo, etc.
- **Satisface criterios de performance**  
 Satisface requerimientos no funcionales (performance, etc.)
- **Cumple expectativas de costo**  
 Para que esto sea posible es necesario planificar, por lo cual se evalúa la relación costo beneficio.

## SCM en el ciclo de vida del software

El software está en cambio constante, y esos cambios pueden ser de tipo internos o externos. Un error, detectado por nosotros o el cliente, porque cambio el negocio o las leyes, etc. Un software cambia siempre durante su uso, si no se usa no cambia. Hay empresas que no tienen claro sus activos de hardware, por lo que también existe la gestión de configuración de hardware.



- **Cambios**
  - **Internos**
    - Correctivo (Defectos)
    - Perfectivo (Mejoras)
  - **Externos**
    - Adaptativos
    - Nuevos requerimientos
    - Cambios en requerimientos

## Problemas con el manejo de Componentes

La administración de configuraciones permite solucionar los siguientes problemas:

- Pérdida de un componente.
- Pérdida de cambios o superposición de cambios.
- Doble mantenimiento
- Cambios no validados.

## Versiones, variantes, reléase

---

### Versión

Una **versión** se define como una instancia de un ítem de configuración que difiere de otras instancias del mismo ítem. El **control de versiones** combina procedimientos y herramientas para gestionar las versiones de los objetos de configuración creados durante el proceso de sw.

"La gestión de configuración permite a un usuario especificar configuraciones alternativas del sistema de software mediante la selección de las versiones adecuadas"; esto se puede gestionar asociando atributos a cada versión (que pueden ser datos sencillos como un nro. de versión asociado a cada objeto. Cada versión de sw es una colección de elementos de configuración (ECS), como ser código fuente, documentos, datos).

### Variante

Si la diferencia que existe entre las instancias de un mismo ítem de configuración es muy pequeña, no se denomina versión sino que se la conoce como **variante**.

### Construcción del Sistema

Es el proceso de compilar y vincular los componentes del software en un programa que se ejecuta en una configuración particular.

### Release

Entrega de un sistema que se libera para su uso a los clientes u otros usuarios de la organización

### La configuración

Una configuración es el conjunto de todos los componentes fuentes que son compilados, sus documentos y la información de la estructura que definen una versión del producto a entregar. La configuración de un software es la sumatoria de todos los ítems de configuración que tiene en un momento determinado, equivale a una instantánea o una foto de todos los ítems de configuración con su versión en un momento del tiempo.

## Planificación de la Gestión de Configuración de Software

---

### Planificación de SCM

Un plan describe los estándares y procedimientos utilizados para la gestión de la configuración. El plan de gestión de configuraciones contiene:

- Tipos de Documentos
- Esquemas de Nombrado

- Estructura del repositorio
- Líneas Base y sus Responsables
- Responsables de creación de procedimientos
- Registros que deben mantenerse
- Herramientas y el proceso para usarlas
- Procesos de Auditoría, ejecución y registro.
- SCM para software externo (opcional)
- Como se hará el control de cambio y miembros del comité de control de cambio

### Consideraciones

- Debe hacerse en tempranamente
- Se deben definir los documentos que se administrarán
- Todos los productos del proceso deben administrarse

## Actividades relacionadas con la Gestión de Configuración

### Administración del cambio

Hacer seguimiento de las peticiones de cambios al sw por parte de los clientes y desarrolladores, estimar costos y el efecto de realizar dichos cambios.

### Gestión de versiones

Seguimiento de versiones de los distintos componentes del sistema y garantizar que los cambios hechos no interfieren entre sí.

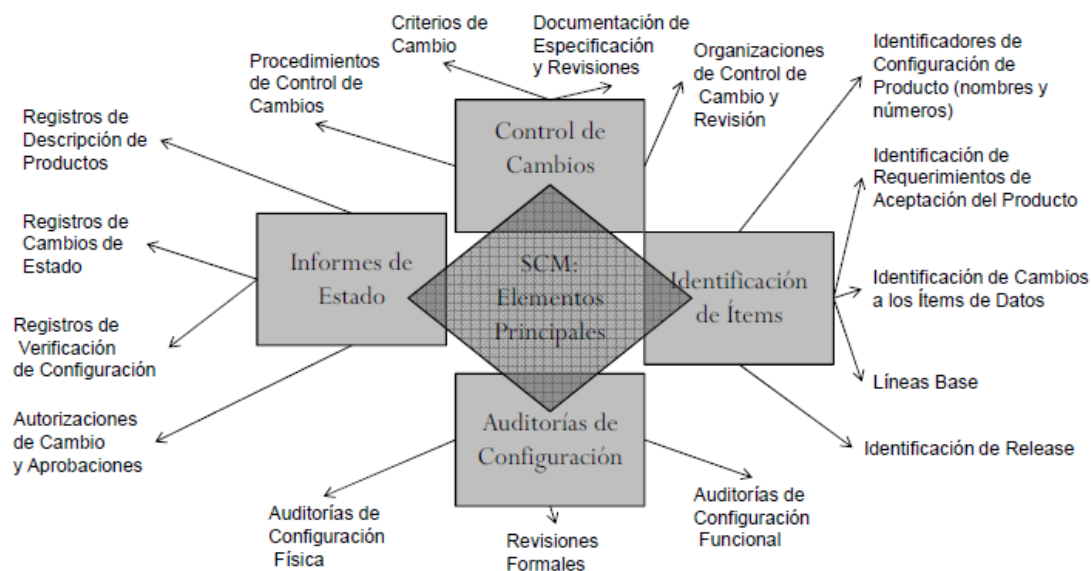
### Construcción del sistema

Ensamblar componentes del programa, datos y librerías, compilarlos y vincularlos para generar un ejecutable.

### Gestión de entregas (release)

Preparar el sw para la entrega externa, hacer un seguimiento de las versiones del sistema que se entregaron al cliente.

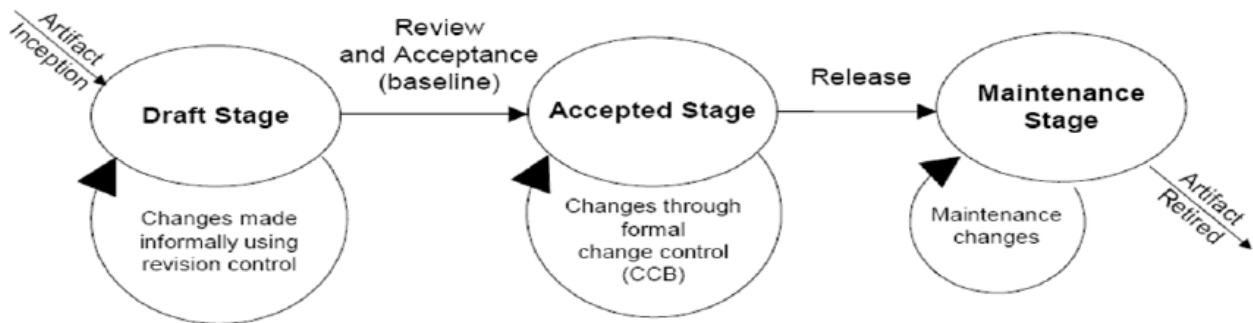
## Elementos de configuración del Software



# Identificación de Objetos en la Configuración de Software

## Identificación de Items de Configuración

Cuando se desea controlar los cambios de un elemento es necesario identificarlo, para lo que se declara un ítem de configuración (SCI). Para que un ítem de configuración se vuelva tal debe estar versionado. Un ítem de configuración inicia en un estado borrador donde se van haciendo los cambios, luego pasa a una etapa de revisión y finalmente queda aceptado. Aquí queda en estado "baseline", es decir, que no se puede cambiar si no es a través de un procedimiento formal de cambio.



## Ítem de Configuración de Software (SCI)

Son todos aquellos elementos que componen toda la información producida como parte del proceso de ingeniería de software, como ser programas de computadora (código fuente y ejecutables), documentos que describen los programas (documentos técnicos o de usuario), datos (de programa o externos).

### Ítems básicos y compuestos

- Objetos básicos: es una "unidad de texto", creada por un ingeniero durante el análisis, diseño, codificación o pruebas. (pe: sección de una especificación de requisitos, listado fuente de un módulo, conjunto de casos de prueba, etc.).
- Objetos compuestos: colección de objetos básicos y de otros objetos compuestos (pe: especificación de diseño).

### Tipos de ítems

- De producto  
Tienen el ciclo de vida más largo, y se mantienen mientras el producto exista. Ejemplo: una ERS, los casos de prueba, la base de datos, el manual de usuario.
- De proyecto  
Los ítems de configuración a nivel de proyecto. El plan de proyecto, el listado de defectos encontrados, tienen un ciclo de vida de proyecto. Un plan de iteración, un burndown chart, se mantiene durante una iteración. La duración impacta también en el esquema de nombrado.

# El rol de las líneas base y su administración.

---

## Línea Base

"Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante, sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambio".

### Características

- Todos los ítems que la componen están revisados y aprobados (por una revisión de pares, por testing, por aceptación del cliente, o por cualquier mecanismo definido previamente).
- Permite repetitividad, permite entregar siempre lo mismo.
- No se modifican ni eliminan ya que se pierde trazabilidad.
- Se identifican con etiquetas, permite encontrar sus elementos a partir de referencias.
- Se almacenan en un repositorio.
- Se definen:
  - Cuando termina un sprint (ágil)
  - Cuando termina cada fase (tradicional)

### Tipos de línea base

- **Línea base de especificación**  
También conocidas como "líneas base de fin de fase", dado que en ellas se definen modelos (requerimientos, análisis, diseño, etc.). Generalmente se definen antes que se tenga código.
- **Línea base operacionales**  
Contiene el producto entregado al cliente, es decir el producto ya ha pasado por un control de calidad definido.

### Identificación de la configuración del Software

Se logra mediante la definición de sus líneas base y de los cambios que los mismos pueden sufrir durante la evolución del sistema.

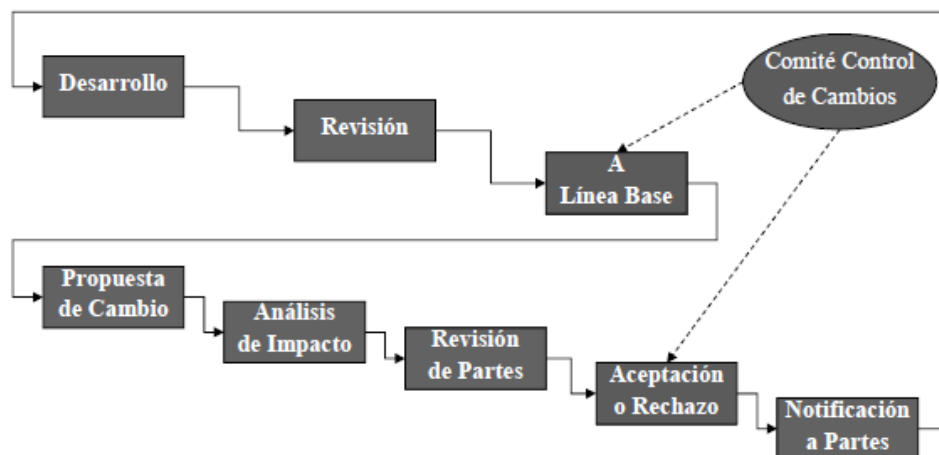
# Control de Cambios

## Proceso de Control de Cambios

Una vez que la línea base se conformó, no es posible cambiarla sin pasar por un proceso formal, llevado a cabo por lo que se conoce como comité de control de cambios. La formalidad del proceso está dada por el hecho de que todos los involucrados se anoticien. Esta autoridad de cambio, al recibir una "**propuesta de cambio**", lleva adelante un **análisis de impacto** del cambio, donde se evalúa el esfuerzo técnico, efectos secundarios, impacto global sobre otras funciones y sobre otros objetos, se le asigna una prioridad para que posteriormente se realice lo que se conoce como **revisión de partes**, en base a todo este análisis el comité **acepta o rechaza el cambio** y **notifica a todas las partes involucradas**.

### Comité de Control de Cambios

- Involucrados directos que están en el proyecto
- Si el cambio se origina en el cliente, se lo incluye
- La convocatoria se realiza para analizar el cambio sobre una línea base, y no un ítem en particular.



# Control de Versiones

## Gestión de versiones y entregas

Es el proceso de identificar y mantener registros de las versiones y entregas de un sistema.

- **Versión**  
Instancia de un sistema que difiere de otras instancias.
- **Variante**  
Si sólo hay pequeñas diferencias entre las versiones, éstas se denominan variantes.
- **Release**  
Una entrega es una versión del sistema que se distribuye a los clientes.



## Identificación de Versiones

Existen tres técnicas básicas utilizadas para la identificación de componentes en una versión particular del sistema.

- **Numeración de las versiones:** al componente se le asigna un número de versión explícito y único.
- **Identificación basada en atributos.**
- **Identificación orientada al cambio.**

## Versionado con herramientas automáticas

Para la gestión de cambios y líneas bases existe herramientas automáticas que brindan una serie de beneficios:

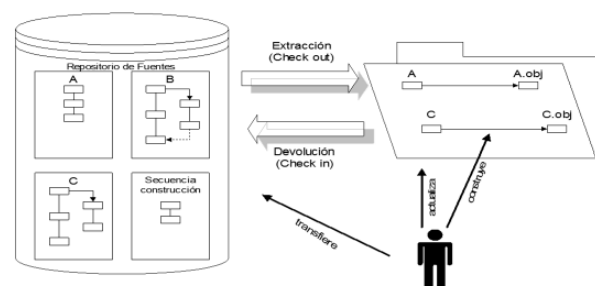
- Fácil acceso a los componentes
- Posibilidad de reconstruir cualquier versión
- Registro de historial de cambio
- Proporciona información resumida.

## Repositorio – Modelo CheckIn/CheckOut

Estas herramientas realizan versionado y trabajan sobre un repositorio. Estos tienen las siguientes características:

- Contienen los ítems de configuración
- Mantienen una estructura de directorios.
- Utilizados para hacer evaluaciones de impacto de los cambios propuestos
- Se implementan mediante una o varias bases de datos.

Para trabajar sobre ese repositorio, se utiliza lo que se conoce como modelo **checkIn/checkOut**. Checkout permite extraer los datos al "Working Area" (esto es la primera vez, luego se hace update), acá es donde se trabaja y se hacen los cambios, al finalizar se hace un Checkin de los mismos al repositorio. Se puede utilizar un esquema de bloqueo completo para evitar inconsistencias, o bien permite la modificación



## Componentes claves a versionar

1. Código fuente
2. Ejecutables
3. Scripts de creación y parámetros de la base de datos
4. Scripts de procedimientos almacenados
5. Scripts de instalación
6. Documentación

# Registro e Informes de Estado de cambios

---

## Reportes del estado de la configuración

Los informes nos dicen el estado actual de la configuración del software, el informe más conocido es el inventario, que contiene una copia del contenido del repositorio.

### Objetivos

- Mantener los registros de la evolución del sistema. Incluye reportes de rastreabilidad de todos los cambios realizados a las líneas base durante el ciclo de vida, es decir nos dice cuáles son las líneas base, cuando se modificó, quien modifico cada cosa.
- Manejan mucha información y salidas por lo que se suele implementar dentro de procesos automáticos.
- Este informe es útil desde el punto de vista administrativo para realizar auditoria. CCB (change control board – comité de control de cambios).

## Registros e informes

- |  |   |
|--|---|
| • Momento de incorporación o actualización de una línea base         | • Estado de la documentación administrativa o técnica           |
| • Estado de cada cambio propuesto                                    | • Deficiencias detectadas durante la auditoría de configuración |
| • Momento en que se incorpora un cambio de configuración de software | • Información descriptiva de cada cambio propuesto              |

## Algunas preguntas que podría responder

- ¿Cuál es el estado del ítem?
- ¿Un requerimiento de cambio ha sido aprobado o rechazado por el CCB?
- ¿Qué versión de ítem implementa un requerimiento de cambio aprobado (saber cuál es el componente que contiene la mejora)?
- ¿Cuál es la diferencia entre una versión y otra dada?
- Causas del reporte de problemas

# Auditoria de Configuración

---

## Auditoría de Configuración

La auditoría de configuración tiene como objetivo asegurar que lo que está indicado para cada Ítem de Configuración de Software en la línea base o actualización se ha alcanzado realmente y que el software y la documentación son internamente consistentes para entregarlos al cliente.

### Debe ser objetiva e independiente

El que controla no debe estar involucrado con lo controlado, debe ser externo del proyecto.

### Funciones

- Determinar la semejanza entre el estado actual del sistema y el establecido como línea base.
- Provee el mecanismo para establecer una línea base.
- Transición desde:
  - línea base a establecer (en etapas formativas),
  - línea base sancionada

## Procesos a los que sirve

### Validación

#### Construir el producto correcto

Asegurar que el problema se ha resuelto de la manera apropiada de tal manera de permitir que el usuario obtenga el producto correcto.

### Verificación

#### Construir el producto correctamente

Asegura que un producto cumple con los objetivos definidos en la documentación de líneas base. Todas la funciones son llevadas a cabo con éxito y los test cases tengan status "ok" o bien consten como "problemas reportados" en la nota de release.

## Tipos de auditorías

### Auditoria de configuración física

Asegura que lo que está indicado para cada Ítem de Configuración de Software en la línea base o actualización se ha alcanzado realmente y que el software y la documentación son internamente consistentes para entregarlos al cliente.

- El objetivo es **verificar**
- Se verifica consistencia contra la documentación
- La práctica indica que primero se hace la auditoria física y luego la funcional.

### Auditoria de configuración funcional

Es la evaluación independiente de los productos de software, verificando que la funcionalidad y rendimiento reales de cada ítem sean consistentes con la especificación del requerimiento.

- El objetivo es **validar**
- Se busca asegurar que el producto es lo que realmente el cliente pidió, y se compara contra la especificación de requerimientos.
- Matriz de rastreabilidad para saber qué caso de prueba corresponde a cada requerimiento.



La gestión en ambientes ágiles se trabaja con **integración continua**, un servidor donde se testea automáticamente el código y luego se integra en producción continuamente. Un repositorio, un esquema de nombrado son cosas que si se utiliza en ágiles, mientras que la auditoria y el comité de control de cambios durante el sprint, pueden obviarse.

## Beneficios de la auditoria

- Aumenta la protección contra cambios innecesarios
- Mejora de la visibilidad del estado del proyecto y sus componentes
- Aumenta la auto responsabilidad
- Disminuye los costos por re-trabajos
- Disminuye el tiempo de desarrollo
- Aumenta la calidad
- Suministra visibilidad y rastreabilidad del ciclo de vida del producto de software

## Desventajas

- Quejas
  - Es burocrático
  - Es molesto
  - Se meten con mi trabajo
- La transición es difícil
- No hay compromiso en todos los niveles
- No hay conciencia del problema

## Mejores Prácticas

- Hacer de la Gestión de Configuración el trabajo de todos
- Crear un ambiente y un proceso de ingeniería que permita la Gestión de Configuración
- Definir y documentar el proceso de SCM, luego seleccionar la/las herramientas que le den soporte al proceso.
- El personal de SCM debe contar con Individuos con expertiz técnica para dar soporte al desarrollo y mantenimiento del producto
- Los procedimientos y el Plan de SCM debe realizarse en las etapas iniciales del proyecto.