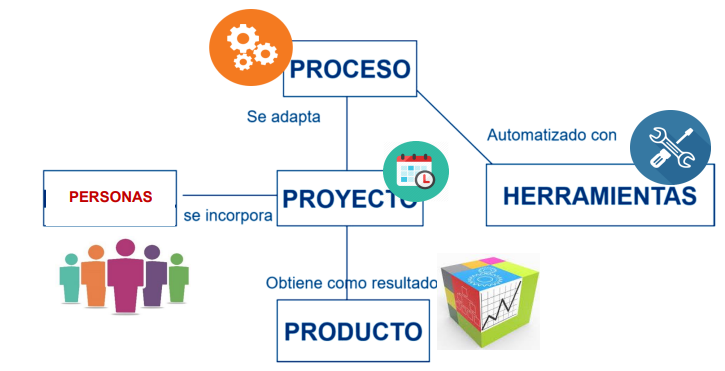
SCM

viernes, 16 de agosto de 2024

10:09

**Software Configuration Management (SCM)**

Es una diciplina de Soporte, que ocurre transversalmente (desde el momento 0 hasta toda la existencia del producto) a lo largo de todo el Proyecto, pero más allá, porque le da soporte al Producto en todo su ciclo de vida. Ósea desde el Primer ITEM DE CONFIGURACION.



Son los ejes por donde la diciplina de Ing de software trabaja.

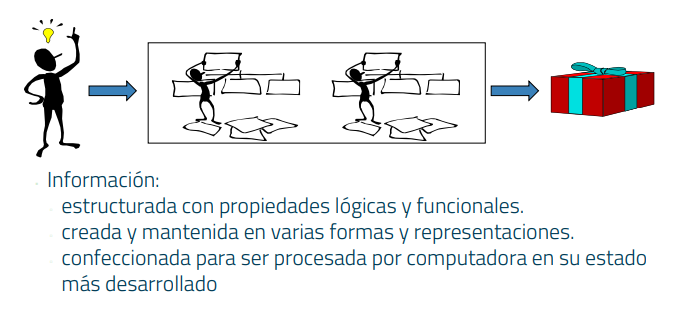
* Producto: Se ocupa del producto como Objetivo final de Calidad
* Proyecto:   
  Es una Unidad de GESTION, es un medio por el cual yo administro los recursos que necesito y las personas que van a estar integradas trabajando en un medio de Gestión para Obtener un PRODUCTO como resultado.  
    
  Es el medio por el cual yo obtengo un Producto de software, ósea que es la Unidad administrativa, la unidad de gestión de la diciplina.  
  Este para poder existir necesita de PERSONAS, RECURSOS Y necesita de un método o Forma de cómo hacer un trabajo para cumplir con el objetivo que se planteó. Ahí entra el PROCESO.  
  Cuando el Proceso entrega esa Definición al Proyecto, este lo va a tomar de acuerdo a lo que le hace falta y de acuerdo a un criterio. Cuando lo toma, arma un Conjunto de tareas que es necesario hacer, conforme al ALCANCE del proyecto, ósea que el alcance se toma de la Definición de PROCESO
  + Alcance se Define como el trabajo que se tiene que hacer para cumplir con el objetivo. Estas Son TAREAS o ACTIVIDADES
* Proceso: Da una definición teórica, de que es lo que hay que hacer para hacer Software. Esta es entregada al Proyecto de acuerdo conforme a lo necesite.
  + Es un Conjunto de actividades que toman como entrada requerimientos y obtienen como salida un Producto o Servicio de software.  
    Este conjunto de actividades están estructuradas por un OBJETIVO, que es lo que yo quiero conseguir

**¿Qué es un PROYECTO?**

Es una Unidad de GESTION, es un medio por el cual yo administro los recursos que necesito y las personas que van a estar integradas trabajando en un medio de Gestión para Obtener un PRODUCTO como resultado.  
  
El Proyecto inicia eligiendo y decidiendo la gente que va a sumarse a trabajar con sus distintos roles en el Proyecto.  
Se encarga de Incorporar PERSONAS que van a Definir Roles. Los ROLES estan definidos en el PROCESO.

**Que es SOFTWARE?**

Es cualquier producto de trabajo que sale de cualquier actividad del ciclo de vida del Proyecto y después del ciclo de vida del Producto.  
Es conocimiento empaquetado a distintos niveles de abstracción.   
Ósea no es solamente el código ejecutable, si no que una base de datos también es Software, una User Story también es Software, es algo más amplio.  
Es cada artefacto que se realizó en un tarea que se realizó en un contexto de un proyecto es software.

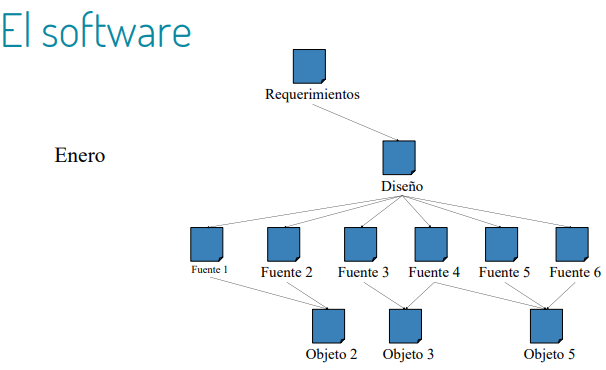


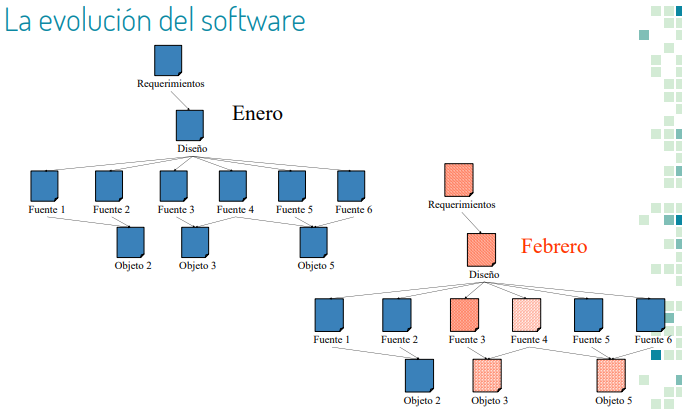
Los ITEMS de Configuración que en general se gestan en el contexto del proyecto van conformando el PRODUCTO

Entonces La DICIPLINA DE GESTION DE CONFIGURACION es la responsable de mantener la integridad del producto. Que esta diciplina de gestión no está encargada solamente una persona, si no que todos tienen la responsabilidad por el producto de software que estamos construyendo, entonces cada vez que alguien crea un ITEM de configuración tiene que respetar los lineamientos definidos respecto de cómo mantener la INTEGRIDAD DELPRODUCTO.  
  
Pero que es INTEGRIDAD DE PRODUCTO?

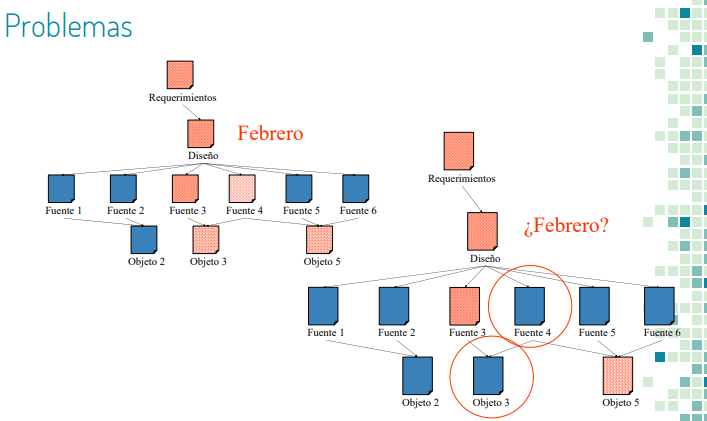
* Satisfacer las Necesidades del USUARIO
* Tiene que ser Fácil y Rastreable durante su ciclo de vida
* Satisfacer los Criterios de la Preforman
* Cumplir con la expectativa de Costo

Cada ITEM DE CONFIGURACION tiene su VERSION:





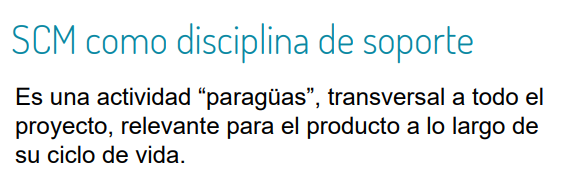
HAY QUE EVITAR QUE NOS PASEN COSAS COMO ESTAS



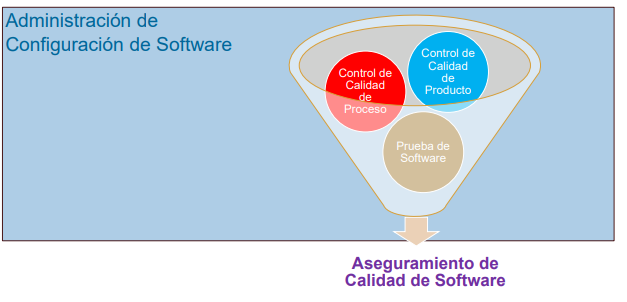
Ósea el no saber cuál es la última versión, donde quedo esa versión o si directamente se eliminó.

Cambios en el Software Tienen su origen en:

* Cambios del negocio y nuevos requerimientos
* Soporte de cambios de productos asociados
* Reorganización de las prioridades de la empresa por crecimiento
* Cambios en el presupuesto
* Defectos encontrados a corregir
* Oportunidades de mejora



**Disciplinas de soporte del Software**



El Marco de trabajo para que otras diciplinas puedan existir, es la:

Administración de Configuración de Software:

* No podemos asegurar la Calidad de un producto que no sabemos dónde está, no sé cuál es la última versión, no se cuáles son los últimos componentes que componen un determinado Relees.
* Esta es la que ayuda que el resto de las Diciplinas puedan funcionar y hacer su trabajo

Definición de Diciplina

Una disciplina que aplica dirección y monitoreo administrativo y técnico a: identificar y documentar las características funcionales y técnicas de los ítems de configuración, controlar los cambios de esas características, registrar y reportar los cambios y su estado de implementación y verificar correspondencia con los requerimientos

¿Por qué deberíamos gestionar la configuración del software?

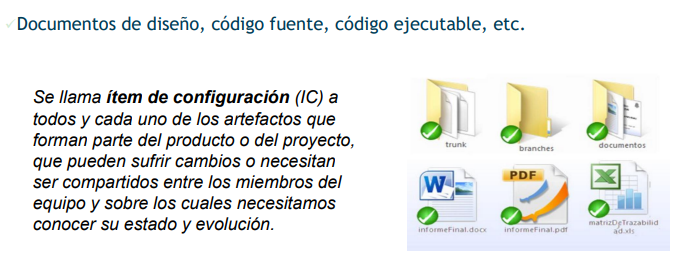
Su propósito es establecer y mantener la integridad de los productos de software a lo largo de su ciclo de vida del PRODUCTO desde que se concibe el producto, hasta que el producto se elimina.

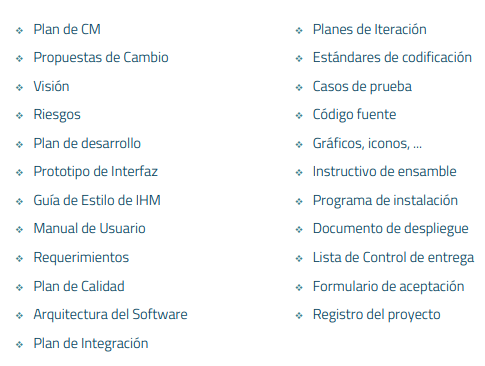
Involucra para la configuración:

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

**Conceptos Clave para la Gestión de Configuración de Software**

* Ítem de Configuración





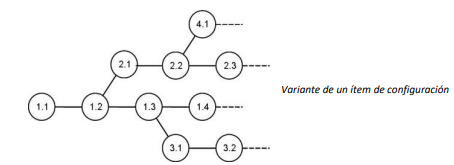
* Versión  
  Una versión se define, desde el punto de vista de la evolución, como la forma particular de un artefacto en un instante o contexto dado.

Es un estado particular en el tiempo de Ese ITEM DE CONFIGURACION.

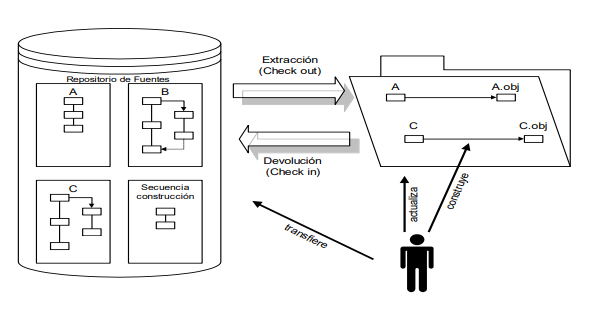
El control de versiones se refiere a la evolución de un único ítem de configuración (IC), o de cada IC por separado.

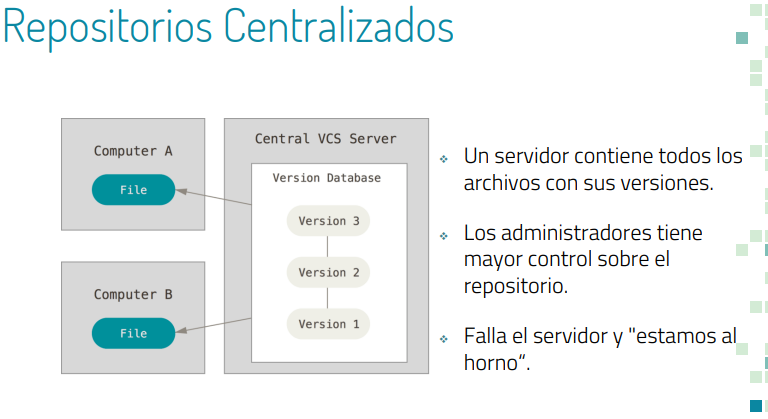
La evolución puede representarse gráficamente en forma de grafo.

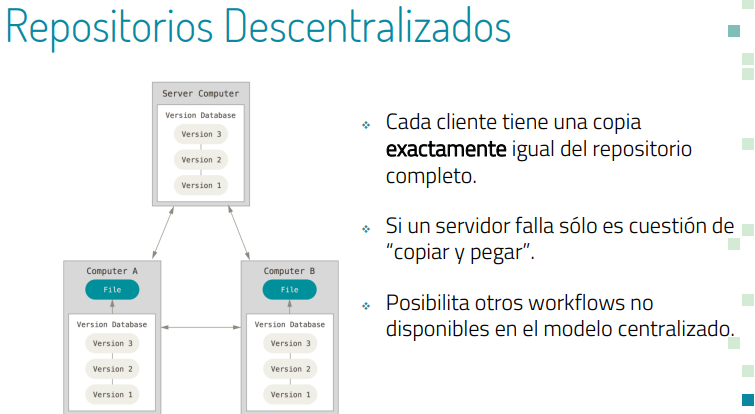
* Variante  
  Una variante es una versión de un ítem de configuración (o de la configuración) que evoluciona por separado.  
  Las variantes representan configuraciones alternativas.  
  Un producto de software puede adoptar distintas formas (configuraciones) dependiendo del lugar donde se instale.  
  Por ejemplo, dependiendo de la plataforma (máquina + S.O.) que la soporta, o de las funciones opcionales que haya de realizar o no.



* ¿Qué es un Repositorio?  
  Un repositorio de información conteniendo los ítems de configuración (ICs)  
  Mantiene la historia de cada IC con sus atributos y relaciones.  
  Usado para hacer evaluaciones de impacto de los cambios propuestos.  
  Pueden ser una o varias bases de datos  
    
  Tiene que tener una estructura ese repositorio ?  
  Si, tiene una estructura que permite definir, que ITEM de Configuración vamos a poner en cada Lugar  
  Esto hace falta para no perder tiempo y encontrar las cosas de forma sencilla y fácil.



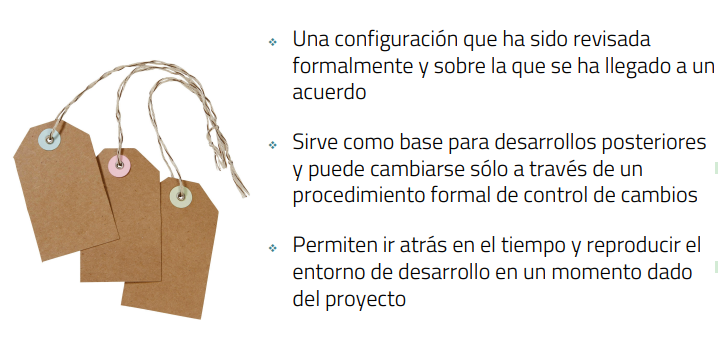


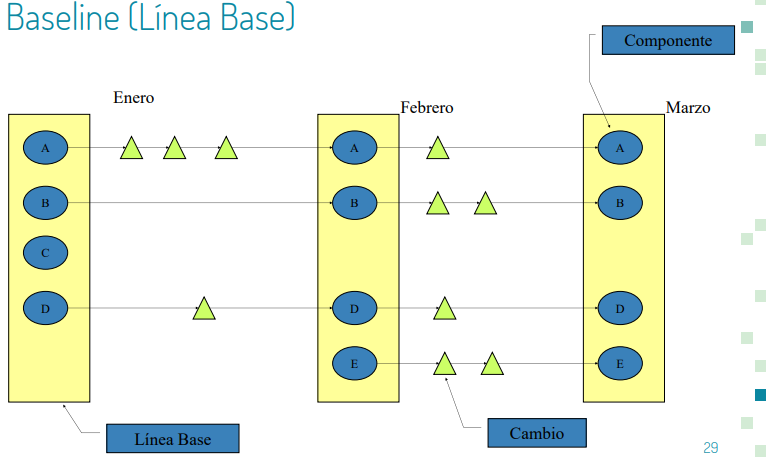


* La Configuración del Software  
  Un conjunto de ítems de configuración con su correspondiente versión en un momento determinado
* Identificación de la Línea Base

Es un ITEM o un conjunto de ITEMs de Configuración con su estado, versión que han sido revisados y gestionados. Y que para cambiarlos deben pasar por un proceso formal de control de cambios

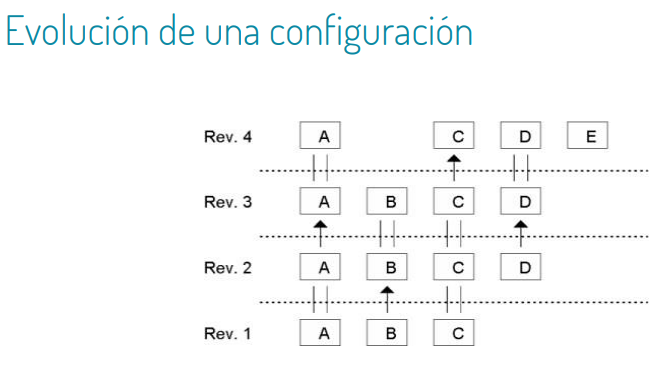
* Se utilizan etiquetas para “marcar” las baseline
* No confundir con la versión del Producto
* Una línea Base puede contener en una línea de tiempo un solo ITEM de configuración, pero es raro, suele tener un conjunto de ítems
* La línea Base tiene que tener un nombre, una versión y una Identificación Única en un momento.
* La línea Base tiene que indicar los ítems de configuración con su estado, que forma parte de esa determinada línea base
* La línea base sirve para para conocer un punto de referencia. La idea de que un algo llegue a ser línea base es porque los ITEM de configuración que están ahí, han sido revisados, probados.





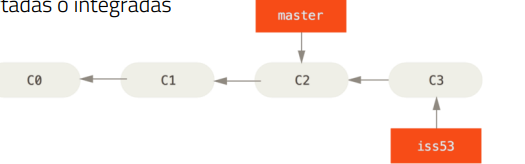
* Representación de Línea Base  
  Hay líneas bases que tiene una sola línea y hay otras que tienen varias líneas bases activas
  + De especificación (Requerimientos, Diseño)  
    Se manejan varias líneas Bases activas, estas no tienen código.
  + Operacionales:  
    De productos que han pasado por un control de calidad definido previamente  
    Tienen Código ya operativo

Si trabajamos con ítems de configuración que no forman parte de la Línea Base, tenemos la libertar de subir y cambiar con toda la libertad. Mientras no forma parte de una línea base, no necesitamos ninguna organización para modificar la misma



**RAMAS**

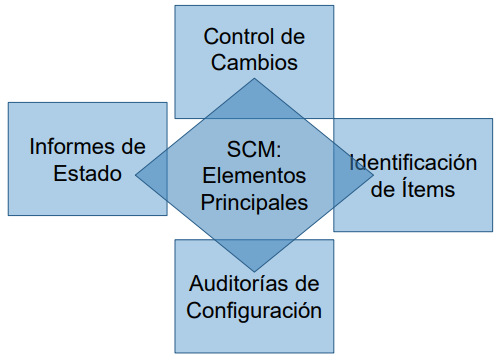
* Existe la rama Principal (Trunk, master)
  + Después se puede hacer una bifurcación de la principal, para trabajar una nueva versión o para revisión.
  + Pueden tener razones de creación con semántica
  + Permiten la experimentación
  + Luego esta puede integrarse o descartarse. Si se integra modifica la rama principal.
    - La operación se llama merge
    - Pueden surgir conflictos (resolvemos con diff)
    - Todas las ramas deberían eventualmente integrarse a la principal o ser descartadas



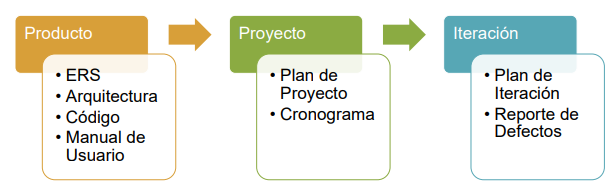
**Definición de Gestión de Configuración de Software**

Una disciplina que aplica dirección y monitoreo administrativo y técnico a: identificar y documentar las características funcionales y técnicas de los ítems de configuración, controlar los cambios de esas características, registrar y reportar los cambios y su estado de implementación y verificar correspondencia con los requerimientos

* **Actividades Fundamentales**



* Identificación de ITEMs  
  En esta actividad va la identificación de ITEMs Unívocamente, Convenciones y reglas de nombrado, estructura de repositorio, ubicación de los ítems en el repositorio  
    
  De acuerdo al ciclo de vida que tenga cada ITEM de configuración, en general en un proyecto de desarrollo de Software, trabajamos con 3 ITEMs de configuración distintos.

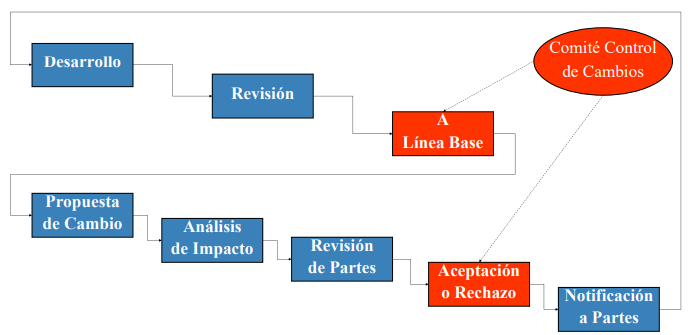


Esto lo hacemos porque cuando vamos a identificar los ITEMs y le vamos a asignar un lugar en el repositorio  
Tengo que tener en cuenta que tipo de ITEM va a ser. Ya que tiene ciclos de vida distintos, el Producto trasciende al Proyecto y el Proyecto a la Iteración.  
  
Una ves que termina la iteración, ya no es relevante mantener los ITEMs de Configuración

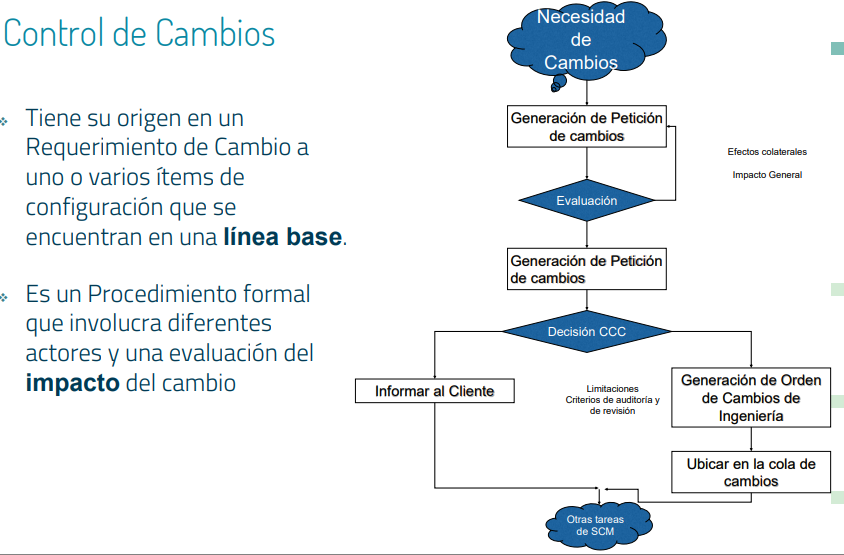
Lo mismo que sucede en un Proyecto

En cambio el Producto, mientras este en producción y halla que darle soporte y este vaya evolucionado. Los ITEMs de configuración de producto van a necesitar una mantención y gestión a lo largo de todo su ciclo de vida.

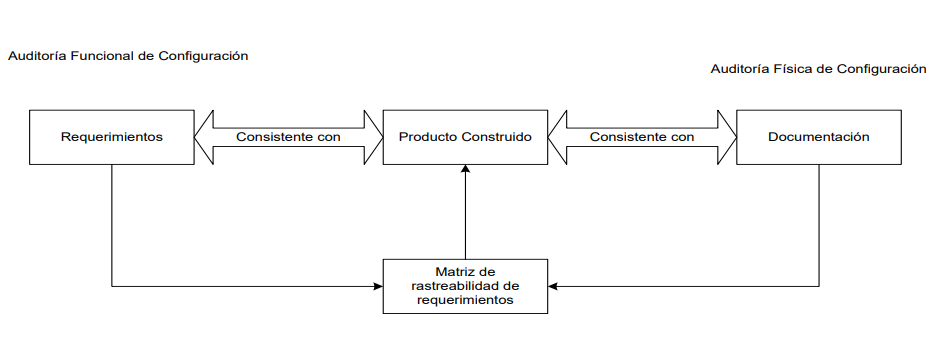
* Control de Cambio  
  Tiene que ver con mantener la integridad en las Líneas Bases  
  Entonces se dice que para cambiar La línea Bases deben pasar por un proceso formal de control de cambios.



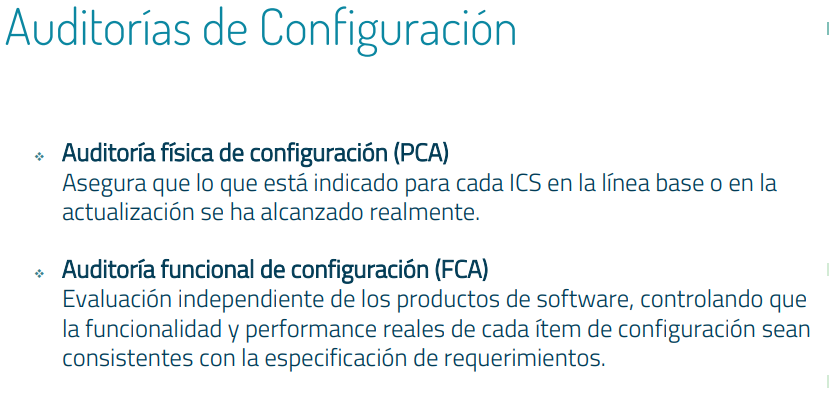
* + Los ITEMs de configuración que no forman parte de una linea base, se los puede cambiar a voluntad y a necesidad. Si forman parte de la Linea Base, se necesita hacer un proceso formal para cambiarlos.
  + El Proceso Formal tiene varias actividades, pero lo primero que hay que hacer es autorizar el cambio y en un segundo momento, hay que verificar que si lo que se cambio es lo que nosotros habíamos autorizado.
  + Quien es el Comité de control de cambio ?  
    Son referentes del equipo, que está trabajando en el desarrollo del producto.
    - Arquitecto
    - El que ve los requerimientos funcionales
    - Desarrolladores
    - Lider de proyecto
    - A veces el cliente tambien
  + Esto se hace asociado a líneas bases,

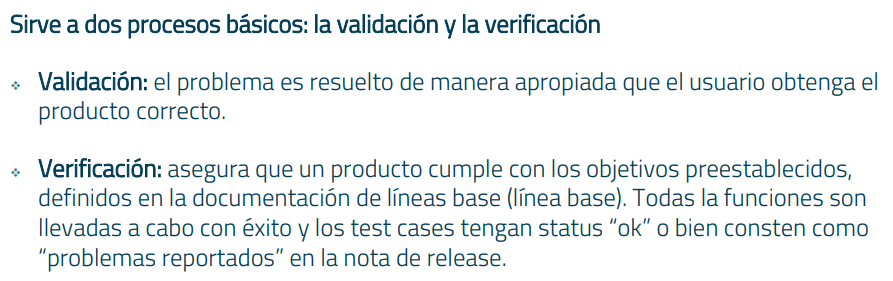


* Auditorias de Configuración de Software

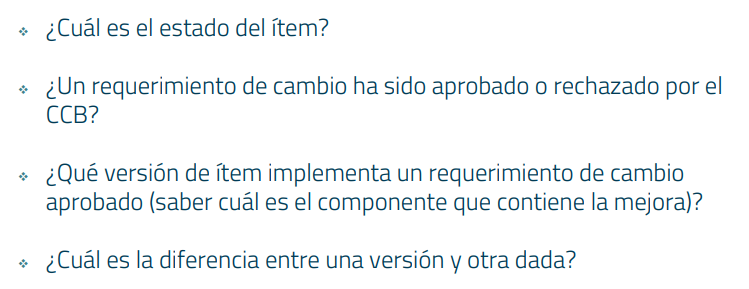


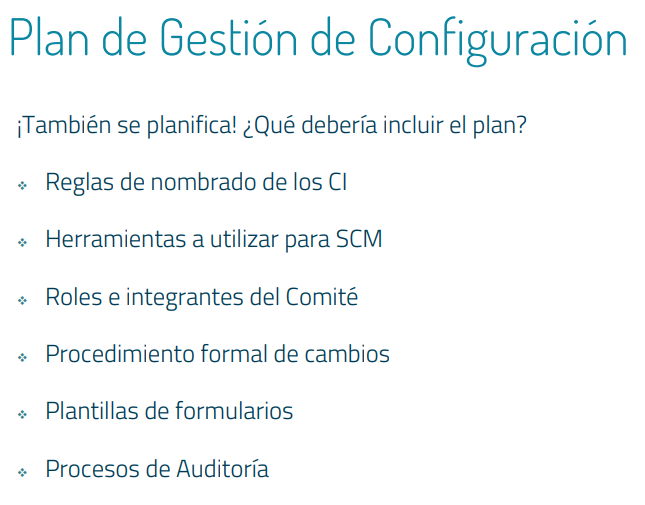
* Hay dos Auditorias:  
  Física (Verificación) y Funcional (Validación)
* Auditoria Física:  
  Vela por la integridad del repositorio (que el repositorio este, que este en el lugar que tiene que estar, que los ITEMs respeten el esquema de nombrado, que estén guardados donde tienen que estar)  
  Se audita una Línea Base  
  El Auditor tiene que ser alguien externo al equipo, porque una auditoria es una revisión independiente.
* Auditoria Funcional  
  Valida si este producto, es el producto correcto. Ósea, si en los ITEMs de configuración que tenemos responden lo que los requerimientos dicen que tiene que hacer.
* Primero se hace la Auditoria Física y después la Funcional
* Las Auditorias necesitan un Plan, si no hay plan, no hay un lugar donde este definida la estructura de repositorio, el esquema de nombrado de los ITEMs, la conformación del comité.  
  Entonces como una auditoria es un proceso de control, si no hay un Plan no tienen que controlar
* Una Auditoria toma un plan y controla una Línea Base y en función de ese proceso de control, se obtiene un Informe de auditoría que muestra todas las desviaciones que se detectan en el momento que se hace.





* Informes de Estado
  + El propósito de esta actividad es hacer reporte, generar visibilidad para la toma de decisiones.
  + Se ocupa de mantener los registros de la evolución del sistema.
  + Maneja mucha información y salidas por lo que se suele implementar dentro de procesos automáticos.
  + Incluye reportes de rastreabilidad de todos los cambios realizados a las líneas base durante el ciclo de vida.
  + Algunas Preguntas que Podria Responder:



* 

**Gestión de Configuración de Software en ambientes Ágiles**

Es una diciplina que es fundacional para lo de la integridad de producto y que ha permitido que los equipos evolucionen en su forma de construcción de software y que traten de hacer un proceso que sea cada vez más productivo, automatizando.

La actividad que más se resiste dentro de la gestión de configuración de software, es la Auditoria. Ya que el pensamiento Agil apunta a que el equipo es quien resuelve y decide. En cambio la Auditoria es algo Externo

SCM en Agile

* Sirve a los practicantes (equipo de desarrollo) y no viceversa.
* Hace seguimiento y coordina el desarrollo en lugar de controlar a los desarrolladores.
* Responde a los cambios en lugar de tratar de evitarlos
* Esforzarse por ser transparente y "sin fricción", automatizando tanto como sea posible.
* Coordinación y automatización frecuente y rápida.
* Eliminar el desperdicio - no agregar nada más que valor.
* Documentación Lean y Trazabilidad.
* Feedback continuo y visible sobre calidad, estabilidad e integridad