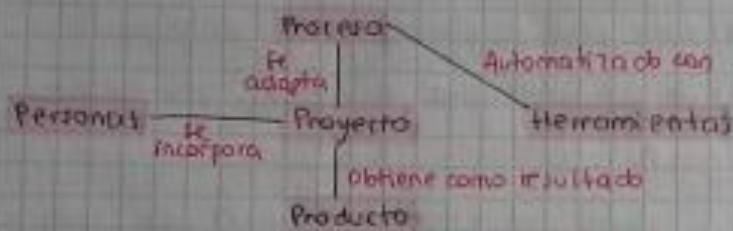


17/04 Software Configuration Management (SCM)

Software en contexto



Tenemos un proceso definido o empírico que se adapta o instancia a n proyectos que toman lineamientos o toman como base ese proceso pero se instancia en cada proyecto único que tienen como resultado un producto de software, donde para poder ejecutarlo se necesita de personas colaboradoras que funcionen entre sí formando un equipo de trabajo para trabajar en esa instancia de proyecto y para obtener ese producto finalmente las herramientas ayudan a que este proceso se automatice lo más posible.

¿Cuándo pensamos en software en qué pensamos?

Conjunto de = programas, procedimientos, reglas, documentación, datos. El software no es solamente el conjunto de programas.

El software

Para poder construir un producto de software se debe seguir un proceso sea definido o empírico, para finalmente poder obtener ^{desde} el planteo inicial del objetivo que tenemos con ese proyecto finalmente la construcción de ese producto de sw, cuya representación puede tener múltiples características o información estructurada con propiedades lógicas y funcionales, creada y mantenida en varias formas y representaciones y confeccionada para ser procesada por computadora en su estado más desarmado.

Ejemplo:

Imagínemos que tenemos un producto de software en donde tenemos los requerimientos, el diseño y el código fuente con los datos persistentes. A medida que pasan los meses no se sabe bien cual es el software y empieza la promesa fundamental que es: resolver la gestión de configuración. Una palabra clave a la hora de definirla es que la gestión de config. es la disciplina que nos permite asegurar y mantener la integridad de nuestro producto, es decir que se pueda identificar en un determinado momento del tiempo si el código fuente que tiene que estar vigente es uno o otro. Todo esto sucede porque el SW cambia.

Cambios en el SW = Tenemos origen en:

- Cambios del negocio y nuevos experimentos
- 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 mejoras

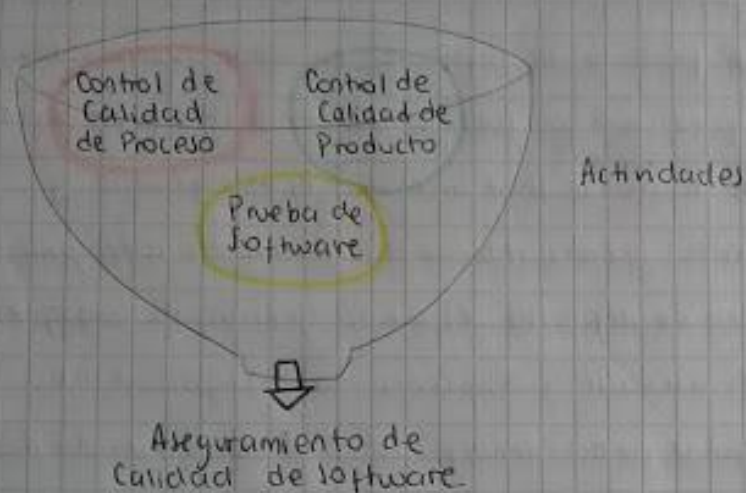
Se m. como disciplina de soporte

Es una actividad llamada "punguaz", es de soporte y transversala todo el proyecto con aplicación en las diferentes disciplinas.

Disciplina de soporte del SW

Administración de configuraciones de SW = se le debe aplicar a lo largo de las actividades de una manera transversal justamente para que la integridad del producto de SW pueda mantenerse. No se hace algo aparte sino que

es propio de incluir estas disciplinas o actividades dentro de las act. de construcción del SW que voy ejecutando a lo largo del proyecto.



Un poco de historia

la gestión de configuración surge a mediados de 1950 cuando CM (configuration management) originalmente utilizado para desarrollo de HW y control de producción, fue utilizado en desarrollo de SW. CM tiene que ver a como hago para encontrar la integridad de mi producto de SW en relación al HW que estoy construyendo.

Definición de ANSI / IEEE 828.

Una disciplina que aplica dirección y monitoreo administrativo y técnico a: identificar y documentar las características funcionales y técnicas de los items 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 debemos gestionar la configuración?

Su propósito es establecer y mantener la integridad de los productos de software a lo largo de su ciclo de vida.

Involucra para la configuración:

Identificarla en el momento dado.

Controlar sistemáticamente sus cambios

Mantener su integridad y origen

Integridad del Producto

Satisface las necesidades del usuario.

puede ser fácil y completamente reentrenado durante su ciclo de vida.

Satisface intenciones de performance

cumple con sus expectativas de costo.

Problemas en el manejo de componentes

Pérdida de un componente

Pérdida de cambios (el componente que tengo no es útil)

Sincronía fuente - objeto - ejecutable

Regresión de fallas

Doble mantenimiento

Superposición de cambios

Cambios no válidos

Algunos conceptos claves para la gestión de configuración de ISW

○ Ítem de configuración: es todos o cada uno de los artefactos que forman parte del producto o del proyecto, que pueden sufrir cambios o necesitan ser compartidos entre los miembros del equipo y sobre los cuales necesitamos conocer su estado y evolución.

Ejemplos: Documentos de diseño, código fuente, código ejecutable, etc.

A lo largo de la vida del producto necesitamos saber cómo éste evoluciona.

Si un artefacto reúne todas las características es un ítem de config.

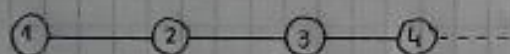
Los items se deben poder identificar antes de crearlos. Cuando se hace el plan de gestión de configuración, ya debería poder identificar cuales son mis items de config. que va a tener mi proyecto

Algunos ejemplos de items de configuración:

- | | | |
|-------------------------|----------------------|---------------------------|
| - Plan de cm | - Plan de desarrollo | - Plan de prueba |
| - Propuestas de cambios | - Manual de usuario | - Código fuente |
| - Visión | - Requerimientos | - Programa de instalación |
| - Riesgos | - Plan de calidad | - Planes de iteración |

VERSION

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



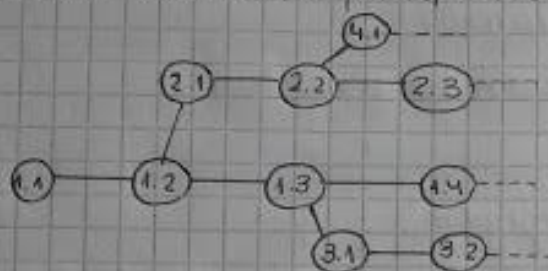
Evolución lineal de un item de configuración.

En un determinado momento la versión del item de config. es la 1, en otro momento del tiempo es la 2, etc.

VARIANTE

- Una variante es una versión de un item de config. que evoluciona por separado.
- Las variantes representan configuraciones alternativas.
- Un producto de software puede adoptar distintas formas (configuraciones) dependiendo del lugar donde se instale.
- Ejemplo: dependiendo de la plataforma (maquina + S.O) que la soporte, o de las funciones opcionales que haya que realizar o no.

Cada variante evolucionará por separado



variante de

un item de configuración.

- Configuración del software: es un conjunto de items de configuración con su correspondiente versión en un momento determinado.
- Línea base = nos va a permitir poder identificar y marcar en un determinado momento del tiempo en etiquetas para saber que en ese momento tengo una configuración que es estable y a la que puedo volver en un determinado momento del tiempo, la cual conserva las características que ya identifique como integridad de mi producto. No sólo alcanza a identificar la configuración del software sino que alcanza a identificar además la línea base.
- Es una configuración que ha sido revisada formalmente y sobre la que se ha llegado a un acuerdo (dejar que eso es una configuración estable)
- Sirve como base para desarrollos posteriores y puede cambiarse sólo a través de un procedimiento formal de control de cambios.
- Permiten ir atrás en el tiempo y reproducir el entorno de desarrollo en un momento dado del proyecto.

Representación de líneas base:

Pueden ser:

- De especificación (Requerimiento, Diseño)
- De productos que han pasado por un control de calidad definido previamente.

○ Ramas = creación de ramas:

- Existe una rama principal (trunk, master) = donde tengo el producto estable
- Sirven para bifurcar el desarrollo
- Pueden tener razones de creación con semántica
- Permiten la experimentación
- Pueden ser descartadas o integradas

Las ramas dan la posibilidad de poder avanzar en las bifurcaciones del desarrollo.

En algún momento se hace un merge si yo quiero integrar esa rama a mi producto.

Integración de ramas

La operación se llama merge, lleva los cambios a la rama principal.

Todas las ramas deberían eventualmente integrarse a la principal o ser descartadas.

○ Repositorio = cuando se arma el plan tenemos que poder identificar los items, que vamos a tener que definir la regla de nombrado, para cada item voy a tener que definir una identificación única porque sino después no voy a poder encontrarlos. Estos items debo almacenarlos en algún lugar, éste almacén será el repositorio.

Basicamente el repositorio es el que guarda cada item con toda su historia y con todo lo que debo ir guardando de ellos.

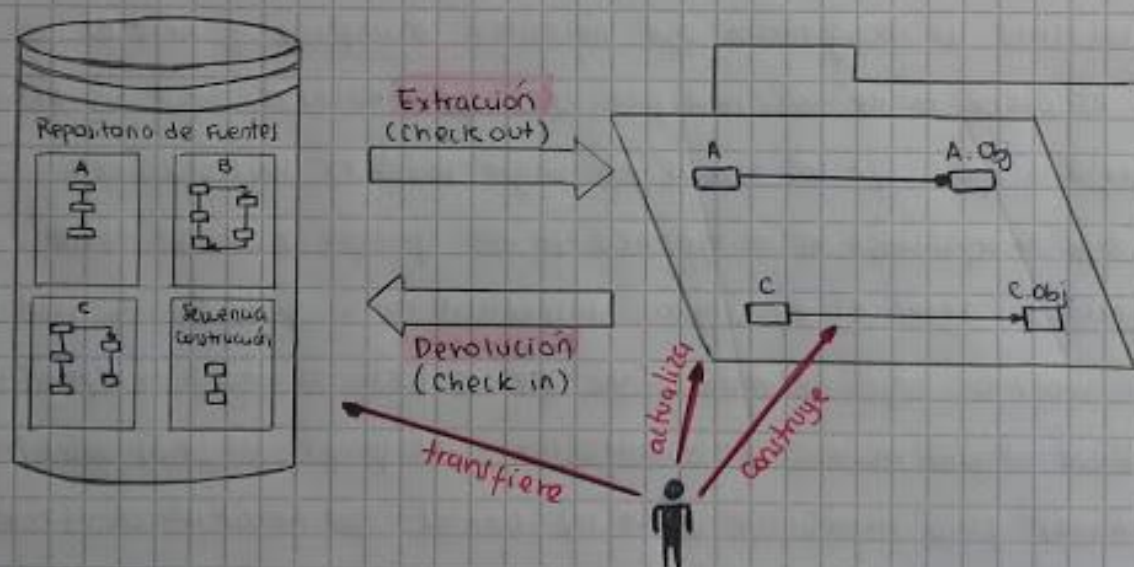
- Un repositorio de información conteniendo los items de configuración (ICs)
- Mantiene la historia de cada IC con sus atributos y relaciones.
- Usado para hacer evaluaciones de impacto de los cambios propuestos
- Pueden ser una o varias bases de datos.

Para ejemplificar un repositorio, debo crear una estructura como de carpetas y dentro de esa estructura voy a tener la construcción por un lado y el almacenamiento de mi repositorio por otro lado. En el medio de esto hay un proceso llamado check in y check out.

La Extracción (check out) es el proceso que le permite a cada persona que está trabajando y desamoliando del producto de JW acceder a lo que está disponible de los ítems de configuración en el repositorio.

La Devolución (check in) tiene que ver con aplicar los cambios que la persona hizo en el ítem de configuración o los nuevos ítems que esta persona creó en el repositorio.

Esto sería el Funcionamiento del repositorio:

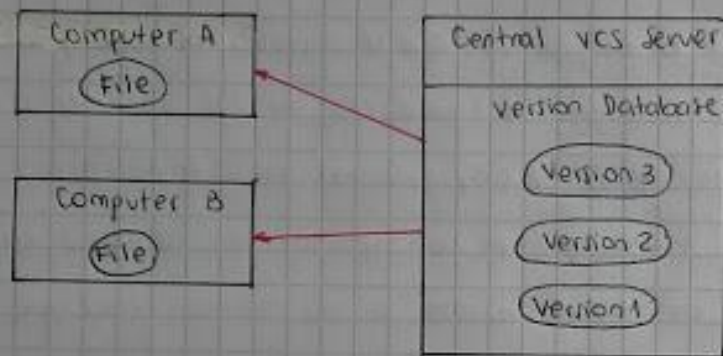


Cuando hay un ítem de configuración que me interesa modificar lo pido a través del check out para poder modificarlo y después lo devuelvo una vez modificado a través del check in. (commit-update)

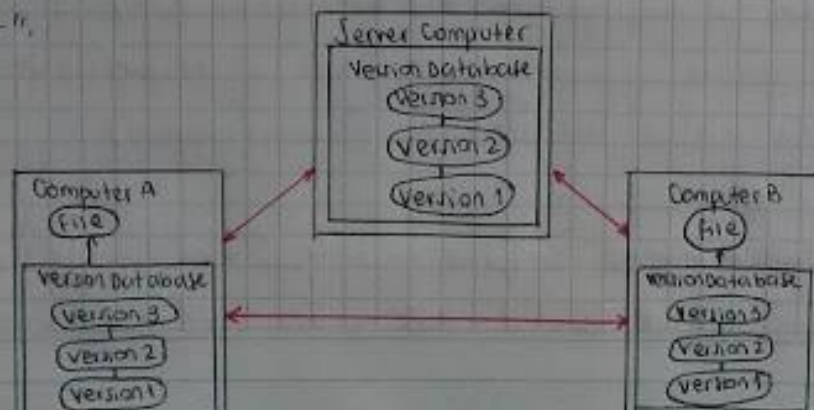
Hay 2 tipos de repositorio:

- REPOSITORIOS CENTRALIZADOS: se tiene en el repositorio todos los archivos y todas las versiones (solo sender) y los administradores

tienen mayor control sobre el repo. y lo que ahí ocurre.
Cuando tiene que ver con check out, se trabaja con un archivo individual
y cuando se termina se sube de nuevo a ese repositorio centralizado.
La desventaja es que si falla el servidor, se tiene un gran problema, ya
que al hacer ese trabajo individual no se lo tiene versionado en el servidor.



• **REPOSITORIO DESCENTRALIZADOS** = en donde cada cliente tiene una copia **exactamente** igual del repositorio completo. Cada uno trabaja con los items de configuración que necesitan modificar y después se hace el check in de las modificaciones que se hicieron sobre la versión del servidor para que los demás en algún momento la actualicen.
Este tipo de repositorio es el que más se usa porque además tiene 2 ventajas: tener el repositorio descentralizado y que cada cliente pueda tener una copia del mismo ya que si falta algo en el repositorio se tiene alguna manera de reconstruirlo, me permite algunos workflows disponibles para interacción entre los clientes que no están en el modelo centralizado. Si el servidor falla solo es cuestión de "copiar y pegar".



Dentro de la Definición de la ANSI/IEEE 829 podemos definir 4 actividades de la:

- Identificar ítems de configuración
- controlar los cambios
- registrar y reportar los cambios y su estado. (actividad automatizada)
- verificar correspondencia con los requerimientos. (auditoría)

Actividades fundamentales de la Administración de Configuración de SW



① Identificación de Ítems de configuración = se deben tener en cuenta varios aspectos =

Identificación única de cada ítem = seña identificarlos de forma única como por ejemplo: si yo defino que un ítem de configuración va a ser un caso de prueba, yo tengo que poder identificar ese caso de prueba de manera única y saber que ese caso de prueba es el correcto, esa identificación única se logra con las convenciones y reglas de nombrado, esto es buscar una regla de nombrado genérica para mis ítems que cuando se la quiera especializar me permita nombrar de manera única cada uno de los ítems. Ejemplo = se quiere nombrar de forma única la especificación de requerimiento, esto es fácil porque cada siempre tendré 1 especificación de requerimientos

de un producto de SW, entonces se la nombrará poniéndole como primera parte el nombre del producto de SW y luego especificación de requerimientos de SW y como habrá una sola en todo mi proyecto el sufijo. Si hay más casos de prueba se deberá poner alguna etiqueta el n° de ca y n° de escenario, y el nombre.

✓ **Convenciones y reglas de nombrado** = se define como nombrar los items para empezar a crearlos, de tal manera que sea único.

✓ **Definición de la Estructura del Repositorio** = en que lugar de la estructura lo voy a guardar a cada item.

✓ **Ubicación dentro de la estructura del repositorio.**

Items de Config. para un proyecto de desarrollo de SW

Producto

- ERS
- Arquitectura
- Código
- manual de usuario



Proyecto

- Plan de Proyecto
- Cronogramas



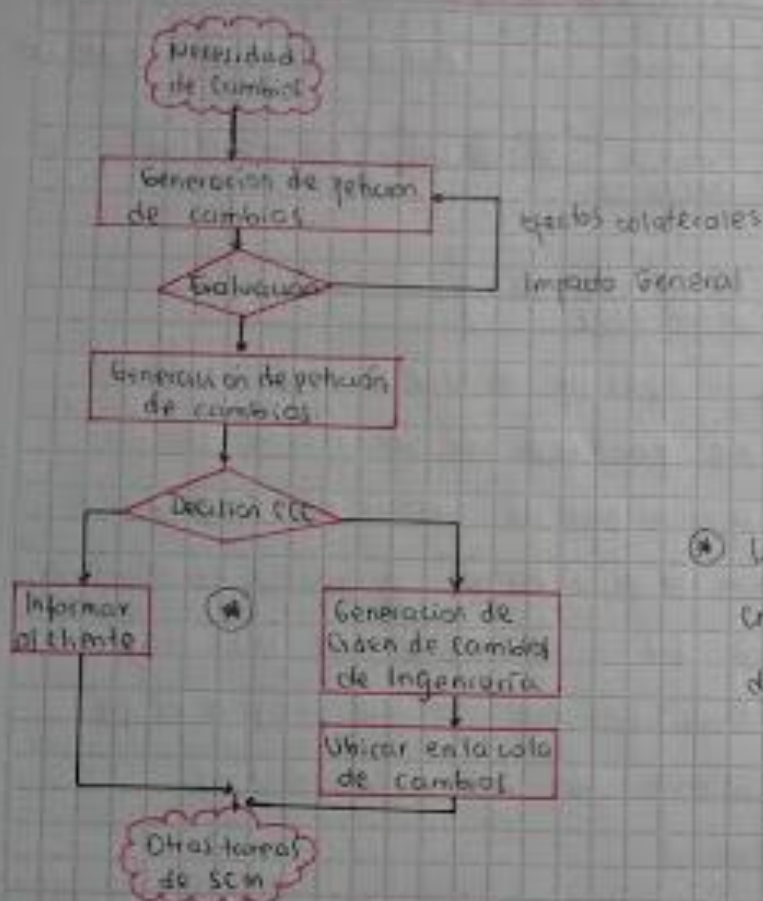
Iteration

- Plan de Iteración
- Reporte de Defectos

② Control de Cambios = cuando queremos modificar una línea base tenemos que someterla a un procedimiento formal de control de cambios. El comité de control de cambios se crea o se define cuando se empieza a trabajar en el proyecto y se lo convoca cuando hay necesidad de realizar algún cambio en los items de configuración que se encuentra en la línea base.

El control de cambios tiene su origen en un Requerimiento de cambio a uno o varios items de configuración que se encuentran en una **línea base**.

Es un procedimiento formal que involucra diferentes actores y una evaluación del **impacto** del cambio.



⊛ Limitaciones

Créditos de auditoría y de revisión

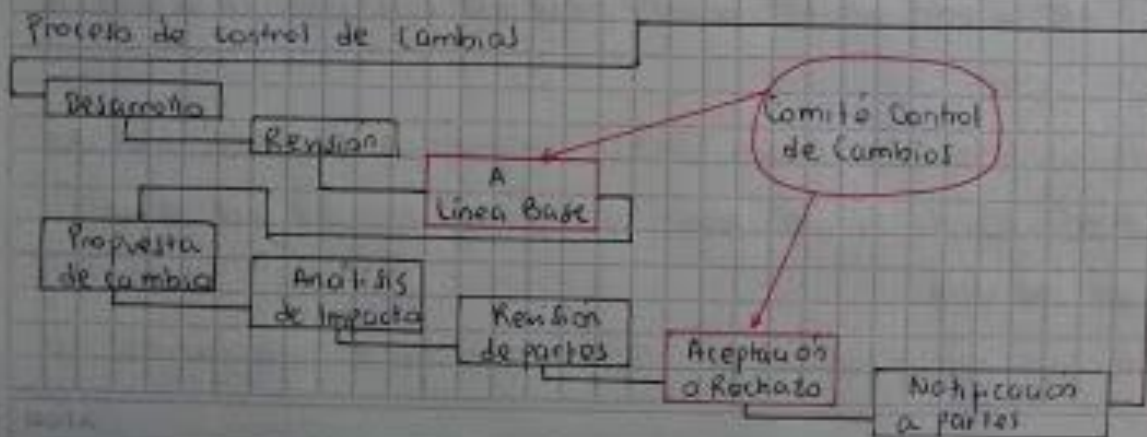
El comité de control de cambios = esta formado por representantes de todas las áreas involucradas en el desarrollo.

Análisis, Diseño

Implementación

Testing

Otros interesados



③ Auditorías de configuración de software = están contradas en 2 aspectos fundamentales: la verificación y la validación

Auditoría Funcional de configuración

Auditoría Física de Configuración



Se tiene el producto construido y 2 tipos de auditorías. Una auditoría lo que hace es verificar que ese producto sea consistente con la documentación, es decir, que si en el plan de gestión de configuración dice que un determinado ítem se va a llamar y guardar de una determinada manera, la auditoría física de configuración debe verificar que eso se cumpla. La auditoría funcional lo que verifica es que el producto construido sea consistente con los requerimientos.

Esto sirve a dos procesos básicos:

- ✓ Validación = el problema es resuelto de manera apropiada que el usuario obtenga el producto correcto, es decir el producto que espera el cliente
- ✓ Verificación = asegura que un producto cumple con los objetivos preestablecidos, definidos en la documentación de línea base (línea base). Todas las 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.

Auditoría física de configuración (PCA) = Asegura que lo que está indicado para cada ICS en la línea base o en la actualización se ha alcanzado realmente.

Auditoría funcional de configuración (FCA) = Evaluación independiente de

los productos de software, controlando que la funcionalidad y performance reales de cada ítem de configuración sean consistentes con la especificación de requerimientos.

④ Informes de Estados = es una actividad fuertemente automatizable porque si se utilizan herramientas para versionar y ver la evolución del sistema, a través de estos informes de estados.

Se ocupan 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 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?

HORA DE UN PLAN!!!

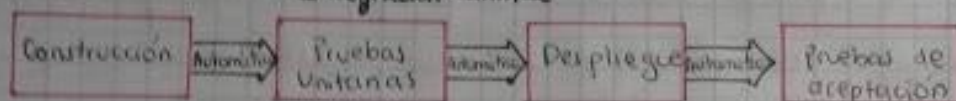
Plan de Gestión de Configuración

También se planifica. ¿Qué debería incluir el plan?

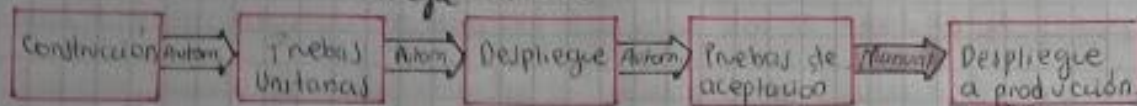
- ✓ Reglas de nombrado de los CI
- ✓ Procesos de auditoría
- ✓ Herramientas a utilizar para SCM
- ✓ Roles e integrantes del Comité
- ✓ Procedimiento formal de cambios
- ✓ Plantillas de formularios.

Evolución de la gestión de configuración de SW

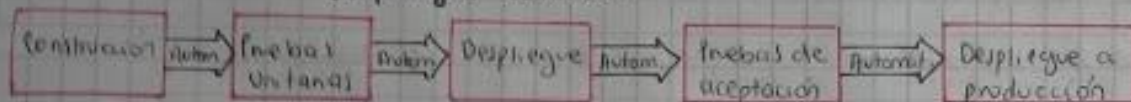
Integración Continua



Entrega Continua



Despliegue Continuo



Gestión de configuración de software en ambientes ágiles

Dentro del Manifiesto Ágil, los valores: la integración de las personas por sobre los procesos definidos y las herramientas, el software funcionando por sobre la documentación, las personas interrelacionadas por sobre contratos y responder al cambio por sobre seguir el plan. Todo esto se va a seguir haciendo no importa en que proceso este. Hay que adaptar las actividades a los principios del Manifiesto Ágil.

En Ágil probablemente no haya actividades de control ni auditorías, etc, pero sí habrá una coordinación ante los cambios.

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.

ISW

TEÓRICO

HOJA N° 26

FECHA

- ✓ Eliminar el desperdicio - no agregar nada más que valor
- ✓ Documentación lean y Trazabilidad
- ✓ Feedback continuo y visible sobre calidad, estabilidad e integridad.

Algunos tips...

- Es responsabilidad de todo el equipo.
- Automatizar lo más posible
- Educar al equipo
- Tareas de SCM embebidas en las demás tareas requeridas para alcanzar el objetivo del Sprint.

08/05

SCRUM