# Unidad 1: Ingeniería de Software en Contexto

Diseño se enfoca en las disciplinas técnicas.

Disciplinas:

* Técnicas -> vinculada al producto de software.
  + Requerimientos.
  + Análisis y desarrollo. -> La materia diseño se centra en este apartado.
  + Implementación.
  + Prueba.
  + Despliegue.
* Gestión -> vinculada al proyecto.
  + Planificación.
  + Monitoreo y control de proyecto (también le dicen, seguimiento y supervisión).
* Soporte o protectoras -> son disciplinas transversales.
  + Aseguramiento de calidad.
  + SCM: gestión de configuración de software.
  + Métricas.

## Software:

Escala de tiempo

Descripción generada automáticamente

El proyecto es el medio por el cual organizamos los recursos, asignar tareas y es el medio por el cual se entrega el producto de software al cliente.

Software es conocimiento, información que se presenta en distintos niveles de abstracción. El nivel de abstracción más alto del software es todo lo que tenga que ver con el negocio.

Software es un conjunto de: programas, procedimientos, reglas, documentación y datos.

La información tiene que estar estructurada con propiedades lógicas y funcionales. Además, es creada y mantenida en varias formas y representaciones. A su vez, es confeccionada para ser procesada por computadora en su estado más desarrollado.

### 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.

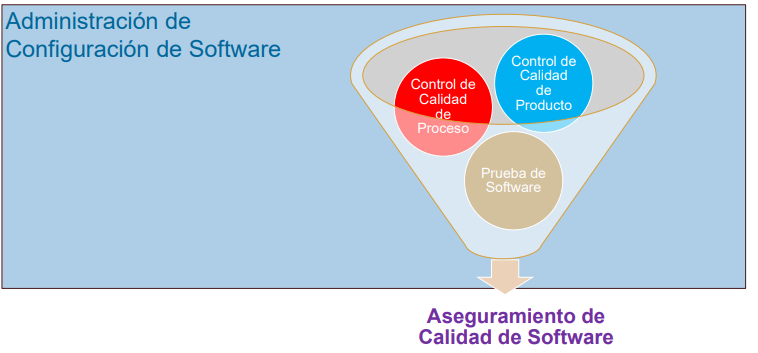
## SCM como disciplina de soporte:

Es una actividad “paraguas”, transversal a todo el proyecto, relevante para el producto a lo largo de su ciclo de vida.

SCM = Gestión de configuración de software.

Con el objetivo de mantener la integridad del producto de software.

### Disciplinas de soporte del software:



### ¿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.

Involucra para la configuración:

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

## Integridad del producto:

* Satisface las necesidades del usuario.
* Puede ser fácil y completamente rastreado durante su ciclo de vida.
* Satisface criterios 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 el último).
* Sincronía fuente – objeto – ejecutable.
* Regresión de fallas.
* Doble mantenimiento.
* Superposición de cambios.
* Cambios no validados.

## Algunos conceptos clave para la gestión de configuración de software:

* Ítem de configuración.
* Repositorio.
* Línea base.
* Ramas (Branch).
* Configuración de software.

### Ítem de configuración de software (SCI):

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

Se llama ítem de configuración (IC) a todos y 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.

Ítem de configuración: es cualquier file system. Cualquier archivo que se puede guardar en un disco y en una computadora. Son cada uno de sus archivos que vamos a administrar. Si se puede meter dentro de un disco duro es un ítem de configuración. Es cualquier archivo que se necesita conocer el estado y su evolución.

### 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.

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 gráfo.

Diagrama

Descripción generada automáticamente

### 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 (configuración) dependiendo del lugar donde se instale.

Por ejemplo, dependiendo de la plataforma (máquina + SO) que la soporta, o de las funciones opcionales que haya de realizar o no.

Diagrama

Descripción generada automáticamente

### Configuración de software:

Un conjunto de ítems de configuración con su correspondiente versión en un momento determinado.

Es la sumatoria de todos los ítems de configuración en un momento de tiempo.

### ¿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.

#### Funcionamiento del repositorio:

Diagrama

Descripción generada automáticamente

#### Repositorios centralizados:

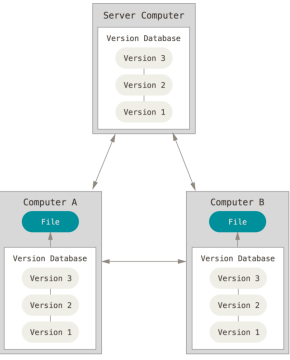
* Un servidor contiene todos los archivos con sus versiones.
* Los administradores tienen mayor control sobre el repositorio.
* Falla el servidor y “estamos al horno”.

Diagrama

Descripción generada automáticamente

#### Repositorios descentralizados:

* Cada cliente tiene una copia exactamente igual del repositorio completo.
* Si un servidor falla sólo es cuestión de “copiar y pegar”.
* Posibilita otros workflows no disponibles en el modelo centralizado.



### Identificación de la línea base:

* Se utilizan etiquetas para “marcar” las baseline.
* No confundir con la versión del producto.

#### Líneas base:

* Una configuración que ha sido revisada formalmente y sobre la que se ha llegado a un acuerdo. Serían ítems de configuración aprobados.
* 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.

Diagrama

Descripción generada automáticamente con confianza media

##### Representación de líneas base:

Puede ser:

* De especificación (requerimientos, diseño).
* Operacional: de productos que han pasado por un control de calidad definido previamente.

## Creación de ramas:

* Existe una rama principal (trunk, master).
* Sirve para bifurcar el desarrollo.
* Pueden tener razones de creación con semántica.
* Permiten la experimentación.
* Pueden ser descartadas o integradas.

### Integración de ramas:

* La operación se llama merge.
* Lleva los cambios a la rama principal.
* 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.

Es una disciplina de soporte, disciplina “paraguas”, está enfocada en mantener la integridad del producto.

### Actividades fundamentales de la gestión de configuración de software:

Hay que hacer 4 cosas cuando se quiere hacer gestión de configuración del software:

* Identificación de ítems de configuración. Cada ítem debe ser identificado unívocamente.
* Control de cambios.
* Auditorias.
* Informes de estados.

Escala de tiempo

Descripción generada automáticamente

#### Identificación de ítems de configuración:

* Identificación unívoca de cada ítem de configuración.
* Convenciones y reglas de nombrado.
* Definición de la estructura del repositorio.
* Ubicación dentro de la estructura del repositorio.

##### Ítem de configuración para un proyecto de desarrollo de software:

Diagrama, Texto

Descripción generada automáticamente

#### Control de cambios:

**Proceso de control de cambios:**

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

* Tiene su origen en un requerimiento de cambio a uno o varios ítems de configuración que se encuentran en una línea base.
* Es un procedimiento formal que involucra diferentes actores y una evaluación del impacto del cambio.

Se necesita un comité de control de cambios

Diagrama

Descripción generada automáticamente

##### El comité de control de cambios:

Está formado por representantes de todas las áreas involucradas en el desarrollo:

* Análisis, diseño.
* Implementación.
* Testing.
* Otros interesados.

El comité de control de cambios analiza los cambios de los ítems de la línea de base.

#### Auditorías de configuración de software:

Diagrama

Descripción generada automáticamente

* **Auditoría física de configuración (PCA):** asegura que lo que está indicando por cada IC en la línea base o en la actualización se ha alcanzado realmente.
* **Auditoría funcional de configuración (FCA):** evaluación independiente de los productos de software, controlado que la funcionalidad y performance reales de cada ítem de configuración sean consistentes con la especificación de requerimientos. Tiene que ver con la trazabilidad de los requerimientos. Usa la matriz de rastreabilidad, es una herramienta que se genera y establece los vínculos de información entre todos los ítems de configuración.

La auditoría es una revisión objetiva e independiente. El auditor debe ser externo al grupo, puede ser de la misma empresa, pero de distinto grupo. Se hace una vez se ha llegado a una línea base (la pide el líder del equipo). Se audita una línea base, que es un momento estable del proyecto.

Sino hay plan de gestión de configuración o línea base no se puede hacer una auditoria, porque la auditoria compara contra el plan de gestión de configuración y la línea base.

Si la auditoría física no se hace no se puede hacer la auditoría funcional.

##### Auditoría de gestión de configuración y V&V:

Sirve a dos procesos básicos: la validación y verificación.

* **Validación:** el proceso es resuelto de manera apropiada que el usuario obtenga el producto correcto.
* **Verificación:** asegura que un producto cumple con los objetivos preestablecidos, definidos en la documentación de líneas 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.

#### Informes de estados:

* Se ocupa de mantener los registros de la evaluación 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?

### Plan de gestión de configuración:

¡También se planifica! ¿Qué debería incluir el plan?

* Reglas de nombrado de los CI.
* Herramientas a utilizar para SCM.
* Roles e integrantes del comité.
* Procedimiento formal de cambios.
* Planillas de formularios.
* Procesos de auditoría.

### Gestión de configuración de software en ambientes ágiles:

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 “si 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.

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.

## Procesos definidos vs procesos empíricos:

Existen dos tipos de procesos:

* Procesos definidos: hay un contexto organización, que define la forma de trabajar, y espera que los proyectos que se vayan a llevar adelante sigan con esta forma de trabajar. Lo que buscan es poder repetibilidad, es poder estimar nuevos proyectos a partir de proyectos pasados. Ejemplos, PDU/RUP.
* Procesos empíricos: se basan en la experiencia, pero plantean una forma de generar esa experiencia. Sostiene que la experiencia no es extrapolable, es decir, la experiencia de uno no se transmite a los demás. Plantean que la mejor forma de tomar decisiones es que la realice el que va a hacer el proyecto. Los equipos tienen que funcionar autónomamente. El empirismo trabaja con 3 pilares que son los que determinan que esto pueda funcionar, estos pilares son los que toman el agilismo:
  + Inspección: paro lo que se está haciendo y reviso. Se inspecciona el proceso (se hace con el equipo) y el producto (se hace con el cliente).
  + Adaptación: adaptarse a los cambios detectados en la inspección.
  + Transparencia: la información debe ser visible y disponible para todas las personas implicadas en el proceso.

En la inspección y adaptación es donde se da la iteración, la realimentación es la experiencia.

El proceso se adapta al proyecto.