

Ingeniería y Calidad de Software

Docentes:

Ing. Laura Couero lcouero@gmail.com

Ing. Cecilia Massano ceciliamassano@gmail.com

Ing. Miquel Crespo miquelcrespo@gmail.com

JTP

Ing. Constanza Gerner conzangerner@gmail.com

Ing. Judith Meles jmeles@gmail.com

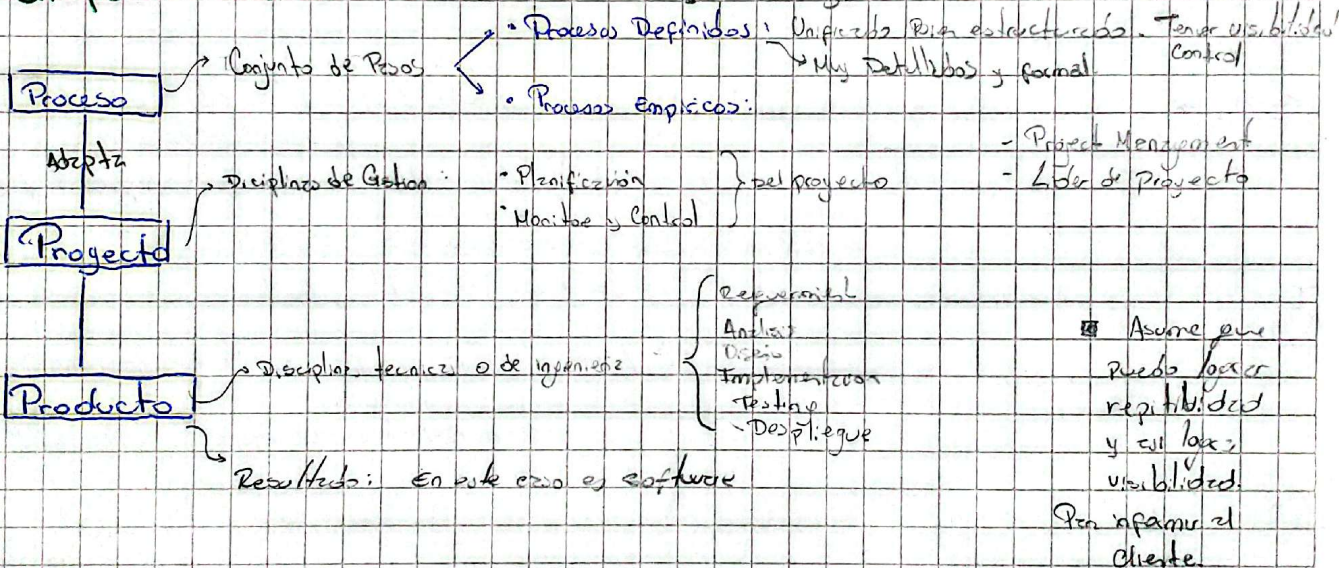
Profesor

Ayudante: Salvador Barbera jsbarbera@gmail.com

Aprobación Directa: Todo nota 7 o superior

No Silver Bullet

Grupos: 8 a 10



Lo 4P

Ingeniería de Software: Disciplina que trabaja desarrollado saberes en las 4 dimensiones

TDD: Forma de desarrollar software. Desarrollo Conductado por Testeo

¿Nota de Negocio? → Que necesita el negocio

- No hay forma de monitorear sin no hay una planificación con la cual comenzar
- Planificar y no monitorear zonas es un desperdicio.

con tener
unos 2 tipos de x

Para que sea
como x

Disciplinas Protectoras (o de soporte)

- ↳ Gestión de Configuración de Software SCM
- ↳ Aseguramiento de Calidad de Proceso y Producto (PPQA)
- ↳ Métricas

Transversales, y a lo largo de
todo el proyecto

Software: Programa o Conjunto de Programas, que aporte conocimiento, y efectos que
aporte valor. Software No es código.

Software es conocimiento/información que se presenta con distintos niveles de
abstracción.

El resultado de cada actividad resulta en un producto de trabajo (entregable)

Todas las actividades:

SCM → es la disciplina que vela por la integridad del producto de software a lo largo de toda su vida

Item de configuración: Cualquier cosa que se pueda poner en un repositorio
→ desde el momento uno hasta el final

Integridad de Software: Disciplina Protectora, transversal a todo el desarrollo del producto

cuyo propósito es asegurar toda la integridad del producto

Integridad → Calidad: es a lo que tiene que ser y lo hace bien (satisface
las necesidades del cliente de la mejor manera).

Hay que preguntarle al cliente que quiere? y pensar en los costos.

R. Funcionales:

R. No Funcionales: Seguro, Rápido, Flexible

❗ Leer paper ❗ No Silver Bullet

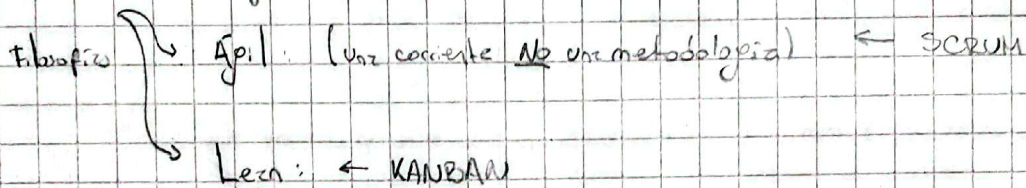
Adapt al proyecto

- Que parte tejo
- Junior, etc
- Como se lleva, se comoda
- Cultura distinta
- Trabaja todo en un mismo sitio

Ej. PUD/RUP/EUP

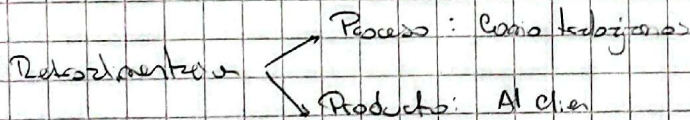
① Defino en tiempo en un proceso definido o empirico. Tomo un modelo de referencia pero todos los adapto.

Procesos Empiricos: Equilibrio entre caos y lo extremo estructurado.



Financiero
Para gestión
↓
No dice nada
de como construir
Software.

Empirico = Experiencia \rightarrow Como se logra, con ciclos de vida cortos por tener más feedback y más rápidos.

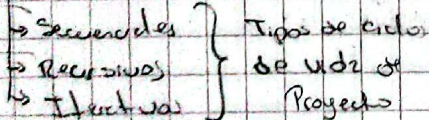


Los procesos empiricos permiten realizar cambios en los procesos sin autorización.

Empiricos \rightarrow Hacen más énfasis en el hacer.

Definidos \rightarrow Hacen énfasis en documentar.

Proyectos y Productos, tienen sus propios ciclos de vida (es decir con distintos)



Con procesos empiricos, solo puedo tener ciclos de vida iterativos

SCM

① Item de Configuración (IC) Contiene archivos que fuer meten en el repositorio.

② Repositorio: Contenedor de todos los items de configuración necesarios.

Deben tener una estructura.

Esto puede definirse en el plan de gestión de configuración de software.

Algunos items sean

- Proyecto nombreMant. <tema>. <ext>
- Producto nombreProducto. Doc. Arg. docx
- Proceso nombreProducto. <nombreComponente>. <ext>

Deben tener nombre único

No para versiones o Fechas (solo poner su ID y luego de versionado)

Minut. Resumen de una reunión, y de los compromisos que quedan acordados.

1 repo x proyecto

③ Lanz Base: un Item de Conf. o un conjunto que tienen madurez, es estable fue validado, testado, etc. Son items configurables

Etiquetas archivos (IC) que conforman la base.

Por cliente
nombreProject. <LanzBase>. Fecha.

Debe ser validado por el comité de Cambios, formado por personas que deben enterarse del cambio

tipo

- Especificación: No tiene código
- Operacional: Tiene código

con su versión.

④ Configuración: Foto de lo que contiene todos los items de configuración que hay en un momento dado (contiene todos los IC del repositorio)