

**ING. DE SOFTW.** (blue box) → **Disciplina q' trabaja desarrollando saberes vinculados a las 4 dimensiones del desarrollo de software (PPPP)**

**Foco de LCS está en:** Disciplinas de Gestión {  
- Planificación  
- Monitoreo y Control  
↳ **Project Manager.**

**DISCIP. PROTECTORAS** (purple box) → **Funcionan como paraguas transversal a todos los procesos, "soportar (dar soporte) a todos"**

- SCM [sw config. management]
- Aseguramiento de calidad de PyP [PPOA] → (U4)
- Métricas

**SCM** (blue box) → **Disciplina protectora transversal cuyo proposito es garantizar integridad del prod de sftw. a lo largo de toda la vida del producto.**

## Procesos Definidos

Permiten repetibilidad y esa rep. me da visibilidad a la hora de hacer un proy. y definir tiempos, costos, prever errores.

Ej: PUD, luego hasta Testing, el RUP agrega todo lo que queda fuera (más allá del Testing).

## Procesos Empíricos

Ej:   
- Ágil ← SCRUM   
- Lean ← KANBAN

Trabaja en ciclos cortos p/ obtener realimentación del prod. y proceso. Permite hacer cambios sin pasar por alguien que lo apruebe. Se enfocan más en el hacer y no tanto en el documentar (como los definidos). Pero sí DOCUMENTAN.

PILARES   
- Transparencia   
- Inspección - Adaptación

Surgen del rechazo de los definidos. Nec. ciclos de feedback RÁPIDO   
Nos permite aprender de la propia XP del equipo, la cual genera trabajando

## SCM

IC (Items de config.)

Repositorio (contenedor de todos los IC q' nec. config.)   
Deben tener una estructura

Linea Base   
[ IC o conjunto de IC, <sup>estables</sup> que están validados y por ende son confiables. Se puede desarrollar en base a esta línea Base.   
Debe ser aprobada x' un Comité de Cambios ]

Configuración de SW

[ "Foto" que se le saca al repo q' contempla q' items tengo, en q' version y en det momento (como una foto instantánea) ]

CCM (Comité control de cambios) → Líder Proyecto, Arquitecto, Analista   
Funcional, Rep. Testing, Rep. Desarrolladores   
- A VECES CLIENTE - (Si C/ muy signif. el cambio)

Para cambios en IC de línea base hay q' llamar al CCM p/ hacer análisis de impacto y analizar el cambio a realizar.

Integridad → Calidad → Cumplir expectativas que el cliente determine necesarias.

26 de Agosto

Ciclo de Vida del Proyecto



Modelo de Proceso

Tipos de CV Ejemplos

- Secuencial → Cascada
- Iterativo → I e I
- Recursivo → Espiral

PROYECTO => Unidad de Gestión de trabajo. Medio x' el cual se org. los recursos y personas que van a interactuar p/ obtener un prod. con valor agregado.

CARACTS:

- Resultado Único
- Elaboración gradual
- Tiene fecha inicio y fin
- Tareas interrelacionadas.

VALORES ÁGILES

VALORAMOS MÁS Individuos e interacciones **POR SOBRE** procesos y herramientas.

- " Software funcionando **POR SOBRE** documentación extensiva
- " Colaborar c/ el cliente **POR SOBRE** Negociar contractual
- " Responder al cambio **POR SOBRE** seguir un plan

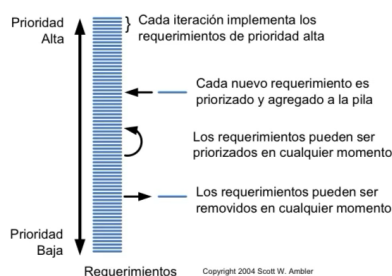
## Los 12 principios del Manifiesto Ágil

- 1 Nuestra mayor prioridad es satisfacer al cliente.
- 2 Aceptar que los requisitos cambien.
- 3 Entregar software funcional frecuentemente.
- 4 Los responsables de negocios, diseñadores y desarrolladores deben trabajar juntos día a día durante el proyecto.
- 5 Desarrollamos proyectos en torno a individuos motivados.
- 6 El método más eficiente de comunicar información es conversaciones cara a cara.
- 7 El software funcionando es la principal medida de éxito.
- 8 Los procesos ágiles promueven el desarrollo sostenible.
- 9 La atención continua a la excelencia técnica y al buen diseño mejor la Agilidad.
- 10 La simplicidad es esencial.
- 11 Las mejores arquitecturas, requisitos, y diseños emergen de equipos auto-organizados.
- 12 A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo y de acuerdo a esto ajustan su comportamiento.

## Requerimientos ágiles

- lo más importante es entregar valor de negocio (OUTCOME) → Resultado con valor agregado.
- Usar modelos p/ mostrar q' construir
- Det. que es "solo lo suficiente" (mín. necesaria).

## PILA DE PRIORIDAD DE REQ.



## SÍNTESIS

- Los cambios son la única constante.
- Stakeholders: no son todos los que están.
- Siempre se cumple eso de que: "El usuario dice lo que quiere cuando recibe lo que pidió".
- No hay técnicas ni herramientas que sirvan para todos los casos.
- Lo importante no es entregar una salida, un requerimiento, lo importante es entregar, un resultado, una solución de "valor".

