

Contenido

REPASO LINEA BASE	2
CUADRO proceso-proyecto-producto	3
MANIFIESTO AGIL.....	4
Valores.....	4
12 principios.....	4
AGILE.....	5
Gestión ágil de requerimientos – users story	5
Product Backlog.....	5
TIPOS DE REQUERIMIENTOS	5
REQUERIMIENTOS - TRADICIONAL VS AGIL.....	6
GESTION TRADICIONAL	6
GESTION AGIL DE REQUERIMIENTOS.....	6

REPASO LINEA BASE

Es una versión estable.

Subconjunto de la configuración de sw.

Ítem/s de configuración, con su versión, que ha alcanzado un nivel de aprobación, según el ítem. Quiere decir que el ítem está en un estado que lo puedo tomar como referencia.

Es como un punto de partida de los ítems.

La estabilidad depende de cada ítem.

La integración a la línea base es a través de un “procedimiento formal”, significa:

- Definir que tenemos que hacer para agregar algo a la línea base.
- Definir lineamientos para aprobar los cambios.

¿Quién forma parte del commit de gestión de cambios? → el equipo de desarrollo, porque el propósito de los cambios controlados es que todos los que necesitan saber que la LB se modificó, se enteren.

Cada línea base debe ser única → un nombre único

Se deben “taggear” los elementos que conformen esa línea base.

Generalmente la línea base va en el “main”.

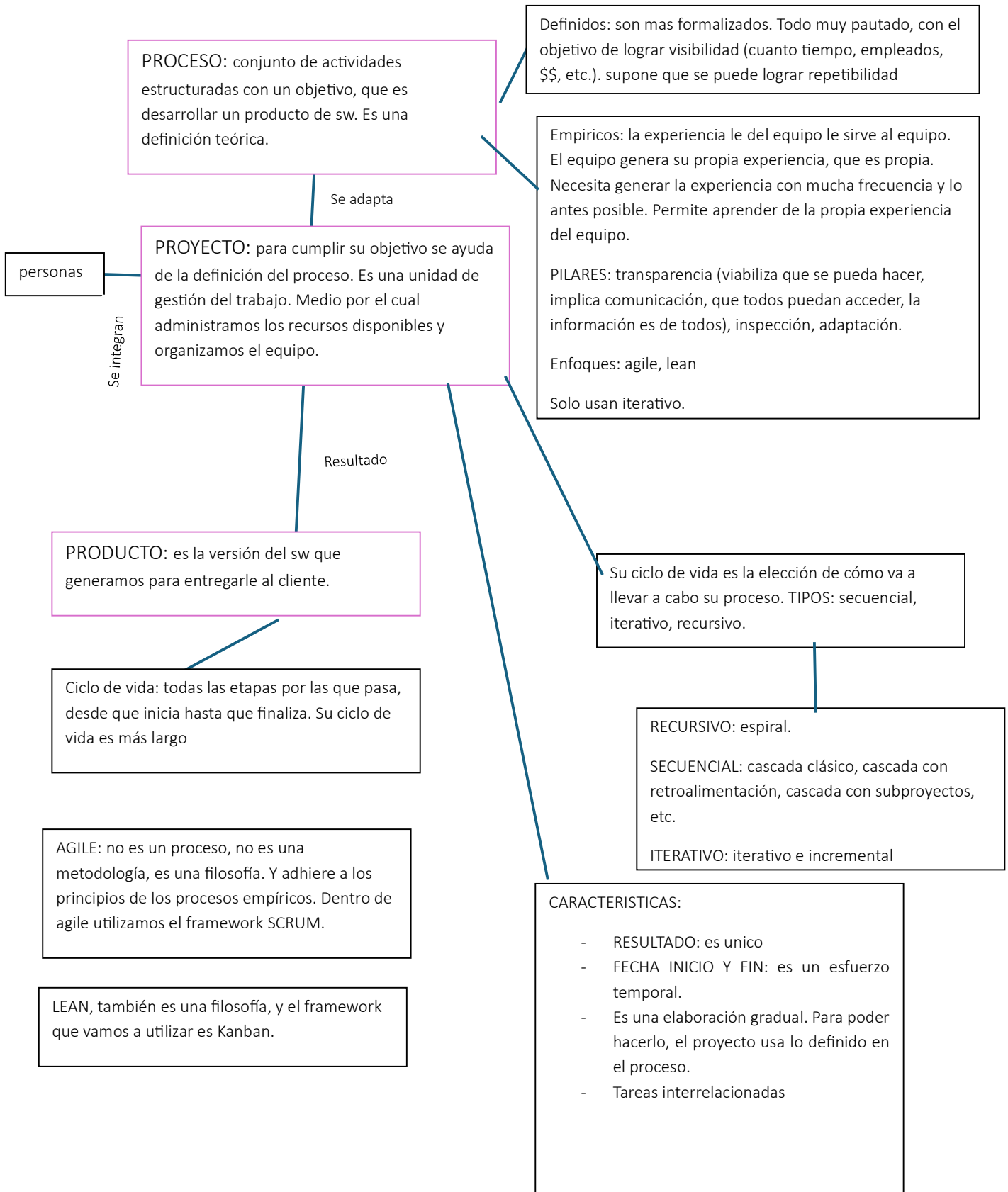
Se audita sobre la LB.

LB de especificación: tiene productos de trabajo varios, pero no código.

LB opracional: contiene código

En el repositorio no hace falta identificar el tipo de línea base.

CUADRO proceso-proyecto-producto



MANIFIESTO AGIL

Valores



Valoramos más a las personas y los vínculos entre las personas, por sobre procesos y herramientas.

Valoramos más SW funcionando, que mucha documentación detallada y extensa.

Valoramos más la colaboración con el cliente, que un contrato firmado.

Valoramos mas estar abiertos a aceptar que los requerimientos cambian, y que van a haber cambios; a seguir un plan escrito.

12 principios

- 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.**

- 1- .
- 2- ..
- 3- .
- 4- El PO sale del negocio. El equipo trabaja en conjunto

AGILE

Es un pensamiento/ideología.

Hagamos un equilibrio entre nada de proceso y demasiado proceso.

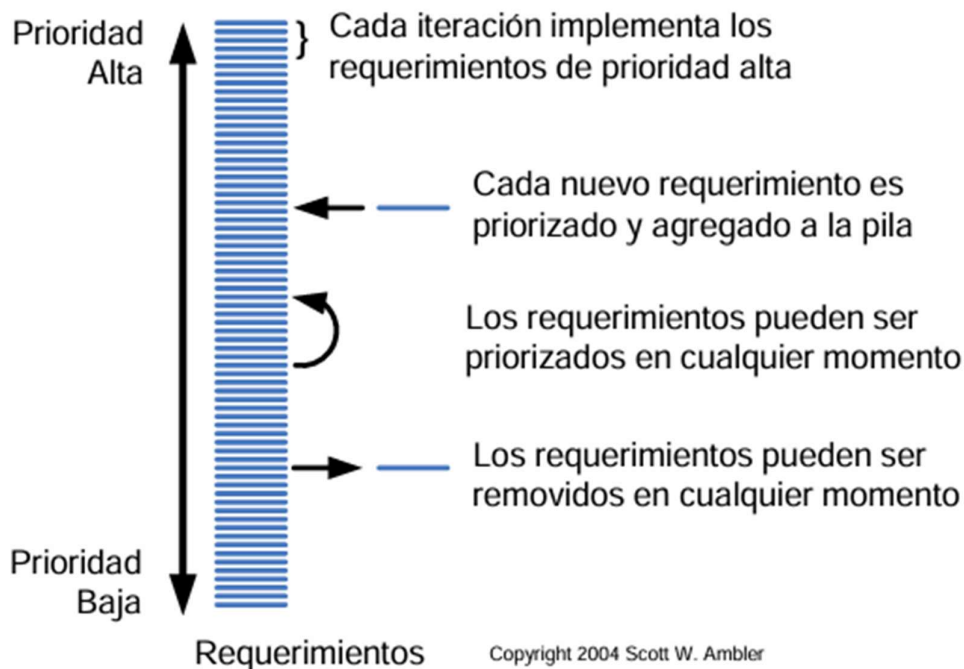
Gestión ágil de requerimientos – users story

Lo mas importante es dejar feliz al cliente con SW funcionando.

Hacemos SW para alguien, ese SW es un medio para cumplir un objetivo.

Lo importante es entregar valor al negocio. Por ello hablamos de resultados y no de salidas. Los resultados/beneficios le dan valor al negocio.

Product Backlog



El PO tiene “poder” sobre esto.

TIPOS DE REQUERIMIENTOS

- De negocio
- De usuario
- Funcional
- No funcional
- De implementacion

REQUERIMIENTOS - TRADICIONAL VS AGIL

GESTION TRADICIONAL

Tengo X cantidad de requisitos y a partir de ello defino la cantidad de recursos y tiempo necesarios.

GESTION AGIL DE REQUERIMIENTOS

Tengo X cantidad de recursos y X cantidad de tiempo, ¿cuánto SW puedo entregar con esos recursos de personal y tiempo?

TRADICIONA;

AGIL

