

## Lo que vas a ver en este post:

1. ¿Qué es el Manifiesto ágil?
2. Los 4 valores del Manifiesto Agile

2.1. Individuos e interacciones por encima de procesos y herramientas

2.2. Software funcionando por encima de documentación exhaustiva

2.3. Colaboración con el cliente por encima de negociación contractual

2.4. Respuesta ante el cambio por encima de seguir un plan

Valor  
Cambio

3. Los 12 principios de Agile

3.1. Satisfacer al cliente mediante la entrega temprana y continua

3.2. Aprovechar el cambio como ventaja competitiva

3.3. Entregar valor frecuentemente  $\rightarrow$  Se Pueden poner en Prod.

3.4. Cooperación negocio-desarrolladores durante todo el proyecto

3.5. Construir proyectos en torno a individuos motivados

3.6. Utilizar la comunicación cara a cara

3.7. Software funcionando como medida de progreso

3.8. Promover y mantener un desarrollo sostenible  $\rightarrow$  Ritmo Sostenible

3.9. La excelencia técnica mejora la agilidad

3.10. La simplicidad es fundamental

3.11. Equipos auto-organizados para generar más valor

3.12. Reflexión y ajustes frecuentes del trabajo de los equipos

## Filosofía Agile Manifiesto Agile

Cuestiones conceptuales es lo que se evalúa en finales

## Desarrollo Ágil de SW (Agile)

Un compromiso útil entre modo de proceso y demanda proceso.

Procesos Agile se basa en procesos empíricos

## Pilares de los Procesos Empíricos

Inspección

Adaptación  $\rightarrow$  se corresponde con.

④ transparencia

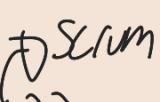
Valores Agiles.

④ RES Ponder al cambio.

La transparencia es uno de los pilares de los procesos agiles. El conocimiento se genera y se mejoran procesos y productos desde este base.

4) Valores y 12 principios

- Los principios visibilizan el cumplimiento de los valores.   
Principios.

De muy importante fuerza y relevancia dentro del negocio } fue lo conocido  
y puede ayudar a la toma de <sup>decisiones</sup> 

4) tecnicos y No tecnicos (RESP. Negocio (PO)) trabajando juntos. Podemos llamar desarrolladores y desarrolladores.

5) El agilismo asume gente comprometida, capacitada, motivada. Si tenemos esa gente deberia funcionar.

6) Comunicación cara a cara es la mejor manera de comunicar si es posible un pizzazón. El 100% apunta a una comunicación efectiva. Lo que uno dice es el 17%. El resto es el 83%.

7) Metrías (a menor forma que mostramos avance es mostrar SW funcionando. Esto afronta contra el desarrollo tradicional. "Mostrame lo que hiciste"

8) desarrollo Sostenible → Estabilidad en el equipo.  
Es complicado en el contexto de la industria.  
Un equipo que trabaja Junto se hace más eficiente.  
Podemos obtener más confiables.

Si no tenemos un equipo estable es muy complicado tener confianza en las estimaciones

9) La calidad del producto NO SE NEGOCIA  
Atención continua en la EXCEPCIONAL TECNICA y el Buen diseño. NO SE CONSTRUYO ALTO y bajo se alta/poca calidad es algo que se construye.

10) La Simplicidad. La mejor solución es la más sencilla. tendremos q complicarnos. Agregar cosas innecesarias  
Simplicidad Maximización del trabajo NO hecho.  
Sino hacemos cosas sin valor

11) Equipos auto-organizados. Arquitecto. Cíder hacen alto y lo bajan al equipo. Se busca cambiar esto y que las arquitecturas, Diseños y requisitos emergen.  
Buscar un equilibrio.

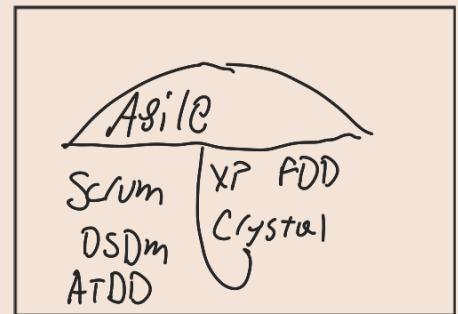
Emerger → surgen y refinen con el paso del tiempo  
Es importante que estén cuando son necesarias

17) Intervalos Regulares (reflexiones) sobre oportunidades de mejora. No tropezar con la misma piedra

¿Qué es ágil?

No es una Metodología o proceso

→ No es porque no es un proceso completo



los framework ágiles apuntan al proyecto.

diseñar la gestión, como se organiza el equipo no tienen prácticas de ingeniería.

Ágil es + definido en el Manifiesto.

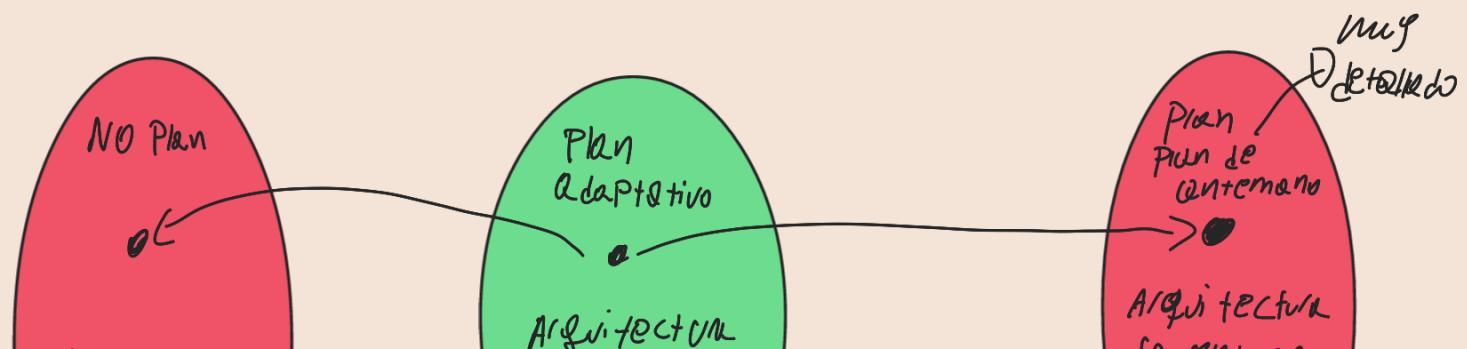
Cambio cultural → sustentando en la arquitectura Ágil de la Org.

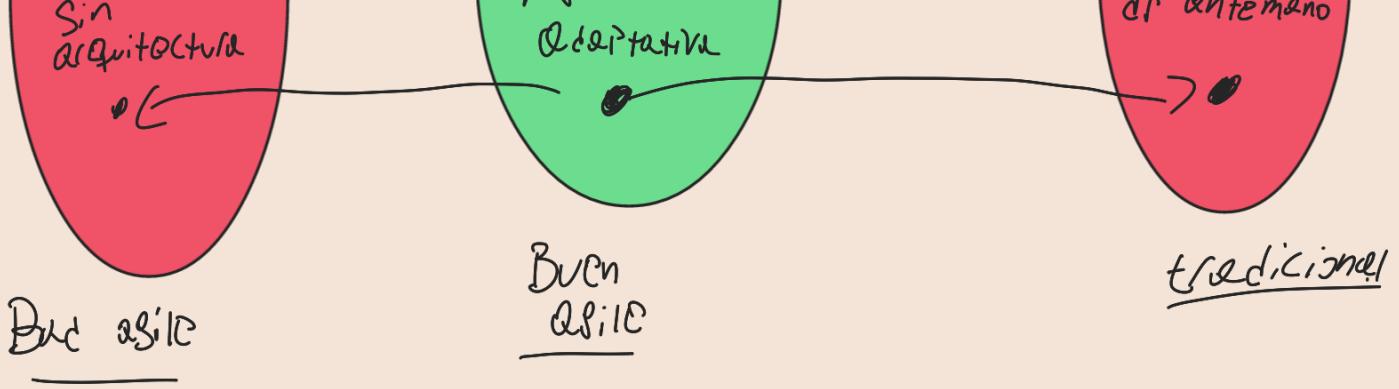
ideología con un conjunto definido de principios y valores que guían el desarrollo del producto.

Técnicas efectivas.

| Daily<br>(sincronización)          | Backlog<br>con<br>prioridades<br>(retropectiva) | Iteraciones<br>cortas | retropectiva | Planeación<br>de<br>interacciones |
|------------------------------------|---|-----------------------|--------------|-----------------------------------|
| → Base<br>para<br>autoalimentación |   |                       |              |                                   |

Ser ágil NO es ser indisciplinado.





ir entregando SW funcional, con menores módulos o funcionalidades pero operativo.

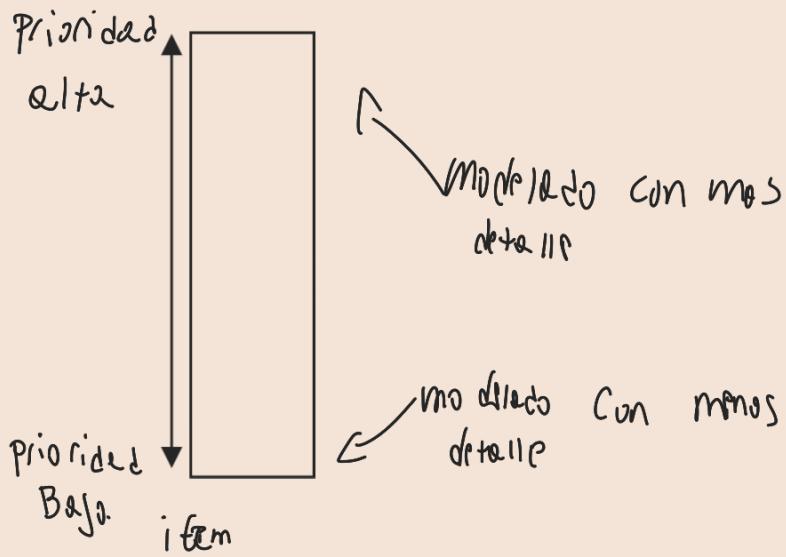
## Requerimientos en Agile

Entregar valor de negocio No solo líneas de código  
hacer más fácil la vida para la organización

Los productos "Exitosos" también tienen un desarrollo significante

80% del valor está en el 20% de la funcionalidad.

Prioridad: Lista priorizada de características del producto



Just in time.

OSVRE para Quitar x falta de MP.

Solo los Req que se necesitan en este iteración son los que serán desarrollados.

Gestión Ágil de Req. → comienzo y trabajo y NO tengo el 100% de los Req totalmente definidos.

El PO es parte del equipo.

"Valor es la obtención de Beneficio tangible o intangible"

+ tradicional → Dirigido x un Plan → Fijo: alcances

Ágil → Dirigido x Valor → Fijo: Recursos y Tiempo.

Lo que tengo claro es una Vision del producto.

No tengo detalles

Niveles de Requerimiento.

Negocio



User → histories de usuarios.



Funcional

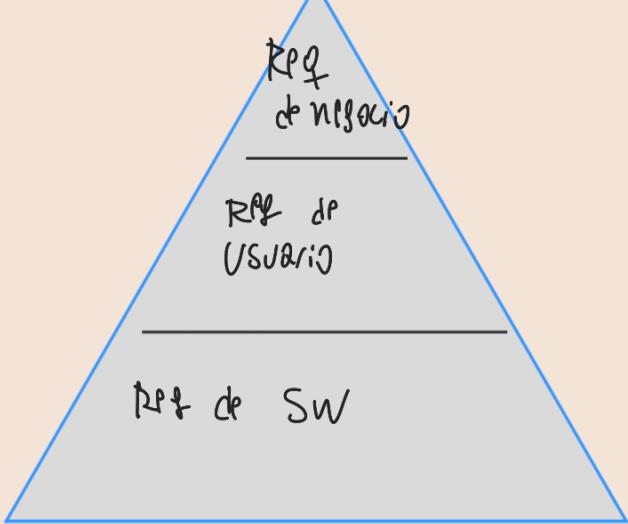


No funcional



Implementación

caso de uso → Req de SW



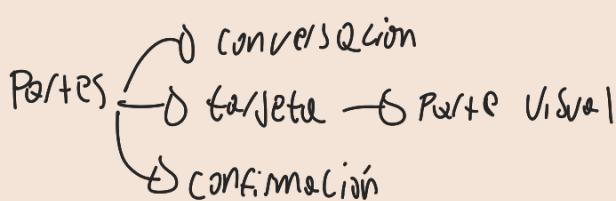
Product Backlog → Especificaciones del Producto

Stakeholders: Interesados e involucrados

"El usuario dice lo que quiere cuando recibe lo que pidió"  
No hay técnicas ni herramientas "Universalis"

---

## User Story



Es una necesidad del usuario. No hay SW

(Card (targeta))

FRASE VERBAL  
(Como <nombre de rol> Yo puedo <actividad> de forma tal  
que <valor de negocio que recibo>  
↳ Para que lo quiere?)



Sí comienza x las User Story de mayor valor de negocio

Sí se agrega para "rastrear", No se usa número (según el autor)

En la realidad si se usa

Dentro la parte superior Izquierda

la parte negativa / la frase verbal <sup>nunca</sup> debe comenzar la User Story

User Story Son Multipropósito

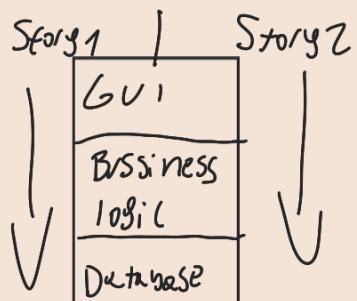
- Una necesidad del usuario - Una descripción del Producto

- Un ítem de Planificación - TOKEN Para conversación

El PO Prioriza las historias en el Product Backlog

Porciones verticales SW Funcionando

Ficha bibliográfica  
aqui se describen  
US y TRU



Roles muy los modelos.  
Diseñarlos, definirlos, especificarlos  
Identificarlos  
Definir la de ROL de USUARIO.

PERSONAS - USUARIO tipo. Se describe un USUARIO para  
Poder nacer inventiva de Producto.

PERSONAJES EXTREMOS. NO son cosas distintas. ACCESIBILIDAD.

USUARIOS REPRESENTANTES (Proxies). Reemplazo al PO cuando  
Este NO es posible. NO son idénticos... hay que EVITAR

Criterios de ACEPTACION → Pueden ser muchos o pocos criterios  
de ACEPTACIÓN

- ↳ definir limites para (US)
- ↳ que tipo ful cumplir
- ↳ Bajar/concretar las cosas de los (US)
- ↳ ayudar a los desarrolladores cuando realiz.

Restricciones → DNF

Criterios de ACEPTACION BUENOS?

DIFINIR una intención, NO una Solución

## (N)dependientes de implementación

los detalles van en las pruebas. Son el resultado de conversaciones con el PO y el equipo

### Prueba de usuario

- Prueba (px perfectiva → Pase; falla)

"User Story Prime" paper en UV.

Definición de lista (DoR)  $\rightarrow$  Ready

Definición de hecho (DoD)  $\rightarrow$  Done

DoR: Sirve para saber que una US que está en el backlog esté lista para ser utilizada en la iteración próxima o actual.

DoD: criterio que establece que una US esté implementada y lista para mostrar al

PO. "terminé mi trabajo", probado, testeado y subido. Checklist

Good vs

INVEST Model

· Indpendiente - Negotiable - Valuable - estimable -  
Small - testable  
Relative.

D) Define que es US

Pueden Entrar.

Madura Poder Estimaciones.

Story Points.

Si una US no tiene  
Estimación es llamada "Spike"

Se busca minimizar Percentage  
de avance y se apunta a Binario (Psta o no)

US

No son CV, No son Paso a Paso

Recurdatorios.

