

## Notas Clase 2 (Filmina de Req Ágiles)

### Req. ágiles:

**Tarea --> Leer en detalle el manifiesto ágil para el viernes 30/08**

### manifiesto.org

#### Procesos definidos vs procesos empiricos:

**Procesos definidos:** Por ejemplo el PUD/RUP, son los que utilizamos en las materias anteriores (ASI). Nos referimos a un contexto organizacional que define una manera de trabajar y espera que todos los proyectos de desarrollo que se van llevando adelante respeten ese proceso que se planteo. Que se establezca todo lo que vamos a necesitar para hacer software. Se realiza una adaptación del proyecto de acuerdo al producto que necesitamos desarrollar. Lo que buscan los procesos definidos es lograr resultados medianamente parecidos a traves de repetir los procesos.

**Procesos empiricos:** Se basan en la experiencia, pero plantean una forma de adquirir esa experiencia a través de un equipo de trabajo. Experiencia = No extrapolable. (Significa que mi experiencia no se la puedo trasladar a ustedes).

Los equipos arman su team building y les sirve al mismo equipo. No les sirve a otros equipos o a nuevos integrantes que reemplazan a integrantes anteriores.

Buscan empoderar al equipo. La mejor forma de tomar las desiciones es quien hace el trabajo. De esta manera se va a ir adquiriendo experiencia.

**El único ciclo de vida que se puede utilizar en procesos empiricos es el ciclo de vida ITERATIVO.**

Tiene **3 pilares fundamentales:**

Se realizan ciclos de inspección y adaptación en donde se adquiere la experiencia. Se inspecciona tanto el producto (con el cliente) y el proceso (con el equipo).

- Inspección: Revisamos el producto con el cliente y vemos cuales cambios hay que realizar. Lo mismo con el proceso para el equipo.
- Adaptación: Se adapta el producto o proceso y se genera experiencia.
- Transparencia: La información debe ser consistente y estar disponible para todos los involucrados. Ser sinceros y hacer visible los riesgos al cliente.

**¿Qué NO es el ágilismo?**

**No es una metodología. No es un proceso. No tiene que ver con la rapidez. Ser ágil no es ser indisciplinado.**

**Es:**

ES UN PENSAMIENTO O UNA FILOSOFÍA. Algo que apunta a la cultura de las personas que lo adoptan. Utiliza procesos empiricos.

### **Manifiesto Ágil:**

Principios (VER)

**Requerimientos ágiles:** Just in time, se comienza con una vision de producto y con un conjunto pequeño de requerimientos y luego se van recolectando los requerimientos cuando los voy necesitando. Durante cada iteracción se implementan los requerimientos de alta prioridad.

### **Users Stories: Requerimientos de software o de negocio. (Ver libro de US del aula virtual)**

Es una descripción corta de una funcionalidad esperada del producto que tiene que satisfacer. Describen una necesidad que viene del product owner.

Pueden tener distintos niveles de granularidad. Si es muy grande se llaman épicas.

### **3 Partes de US:**

- Card: Parte visible de la US que ingresamos en el product backlog, descripción corta. Tiene una sintaxis recomendada:

Como <<Usuario>> (ROL)

Quiero <<Funcionalidades>>

Para <<Valor de negocio>>

Buscar Destino por Dirección	Pruebas de Usuario
<p><b>Como Conductor quiero buscar un destino a partir de una calle y altura para poder llegar al lugar deseado sin perderme.</b></p> <p><b>Criterios de Aceptación:</b></p> <ul style="list-style-type: none"><li>• La altura de la calle es un número.</li><li>• La búsqueda no puede demorar más de 30 segundos.</li></ul>	<ul style="list-style-type: none"><li>❑ Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa).</li><li>❑ Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla).</li><li>❑ Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla).</li><li>❑ Probar buscar un destino en un país inexistente (falla).</li><li>❑ Probar buscar un destino en País existente, ciudad inexistente (falla).</li><li>❑ Probar buscar un destino en un país y ciudad existentes, de una calle existente y demora más de 30 segundos (falla).</li></ul>

- **Conversation:** La parte más importante, es lo que se conversa. Las preguntas, dudas, etc. de lo que se tiene que desarrollar.
- **Confirmation:** Pruebas de aceptación de usuario.

**Spike:** Necesidad de profundizar más sobre lo que me están pidiendo por la incertidumbre, cuando el nivel es muy grande y no le puedo contestar al PO cuanto me voy a demorar, o que necesito. Hay 2 enfoques relacionados a las spikes.

- Vinculadas a cosas del negocio. Ej: AFIP te pide que implementes algo para determinada fecha pero no saben que quieren todavía. El qué hay que hacer es la incognita.
- Vinculada a aspectos técnicos. Ej: Referido a aspectos técnicos en donde el cómo hay que hacerlo es la incognita.

Las tiene que escribir el product owner (las ideas).

**Product backlog:** Lista priorizada de productos, contenedor de items de product backlog. Un item que se puede encontrar es la US. Los items que tienen mayor prioridad estan arriba y hacia abajo de menor prioridad. No puede haber tareas. El product owner es el que arma el product backlog.

Nosotros vamos a utilizar el framework SCRUM. La US tiene que ser consumible en una iteración. Como máximo 1 mes.

I: Independiente

N: Negociable

V: Valor de negocio

E: Estimable

S: Small

T: Testeable