

# ISW 23-8

## Procesos definidos

RUP/PUD.

Parten de una definición, que se establece organizacionalmente (a lo sumo los grupos justifican su desviación de la misma). Tener un proceso estándar  
Entran las 4 p.

Se definen de la forma más completa posible. El proceso se adapta al proyecto.

Lograr predecibilidad.

Se define fuera de los equipos.

## Procesos empíricos

Surgen porque se malinterpretan en las organizaciones. Viene del concepto de experiencia. Lo que plante es aprender de manera constante y sistemática de la propia experiencia del equipo. La experiencia no es extrapolable, es decir, la experiencia generado en un equipo no se puede trasladar a otro equipo.

Los resultados obtenidos son distintos debido a que las experiencias son diferentes. Tiene que ver con el empoderamiento de cada equipo en su autonomía. Busca la autoorganización y autogestión, con la autoridad necesaria para que puedan tomar decisiones.

Necesita un ciclo de vida que permita retroalimentación rápida, por ende deben trabajar con ciclos de vida iterativos e incrementales. Gracias a las iteraciones permito cumplir con los pilares.

## Pilares

### Inspección

A nivel de proceso: Qué tan bien estoy haciendo las cosas?

A nivel de producto: Lo que hice es lo que pidió el cliente? Niveles de satisfacción del producto por parte del cliente.

### Adaptación

Adaptar a cambios

## Transparencia

La información necesaria para trabajar, esta disponible para todos los que intervienen en el proceso.

## Filosofías

Ambas adhieren a los procesos empiricos y deben declarar como vamos adherirnos a los 3 pilares del empirismo.

## Ágil

Es una filosofía. No son metodologías.

SCRUM NO SE PUEDE LLEVAR ADELANTE CON CASCADA.

Estamos hablando en un pensamiento, que busca ofrecer software basandose en los pilares empíricos. Fundadose sobre equipos autogestionables y autónomos, donde la calidad del software no se negocia. Es una filosofía minimalista

## Valores

Valor entregado al cliente, estos conceptos son los que mas entregan valor.

- Valoramos mas los individuos e interacciones sobre el uso de procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colabrar con el cliente sobre negociación contactual. La negociación implica ver quien tiene la culpa, mas que lograr un acuerdo o relación.
- Responder al cambio sobre seguir un plan. No me aferro a un plan, lo diseño pero contemplo que esta sujeto a cambios.

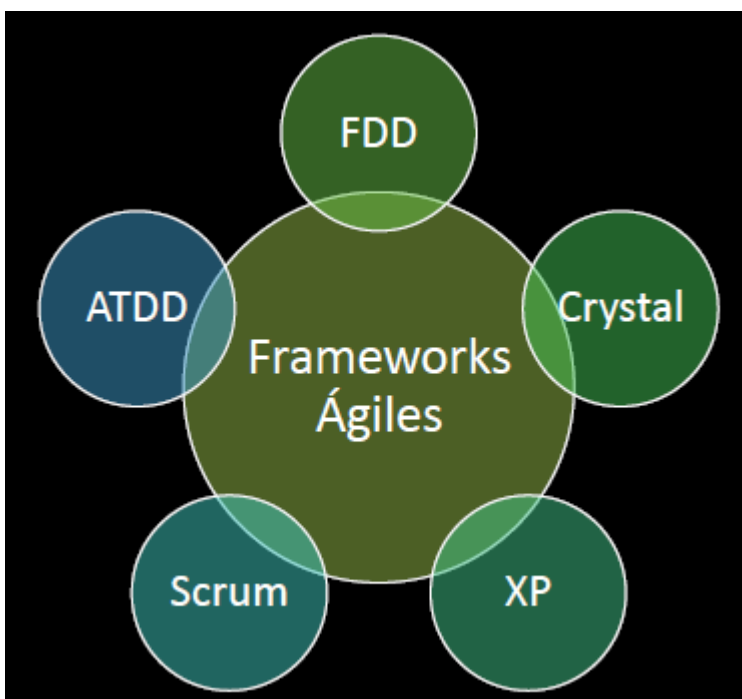
## Principios



4. Los responsables de negocios, diseñadores y desarrolladores deben trabajar juntos día a día durante el proyecto. Personal técnico y no técnico. Le da un nivel de participación y de peso en la toma de decisiones. Donde el no técnico incorporamos al PO, quien decide por el negocio, y participa activamente en los vistos buenos o malos de los releases.

6. El mejor medio de comunicación en el equipo, el más eficiente, es la comunicación cara a cara. Esto se funda en equipos preparados, motivados, que asumen el compromiso.

## Frameworks ágiles



Agilismo no es ser indisciplinado, son pocas reglas que se deben cumplir a

raja tabla.

## Requerimientos ágiles

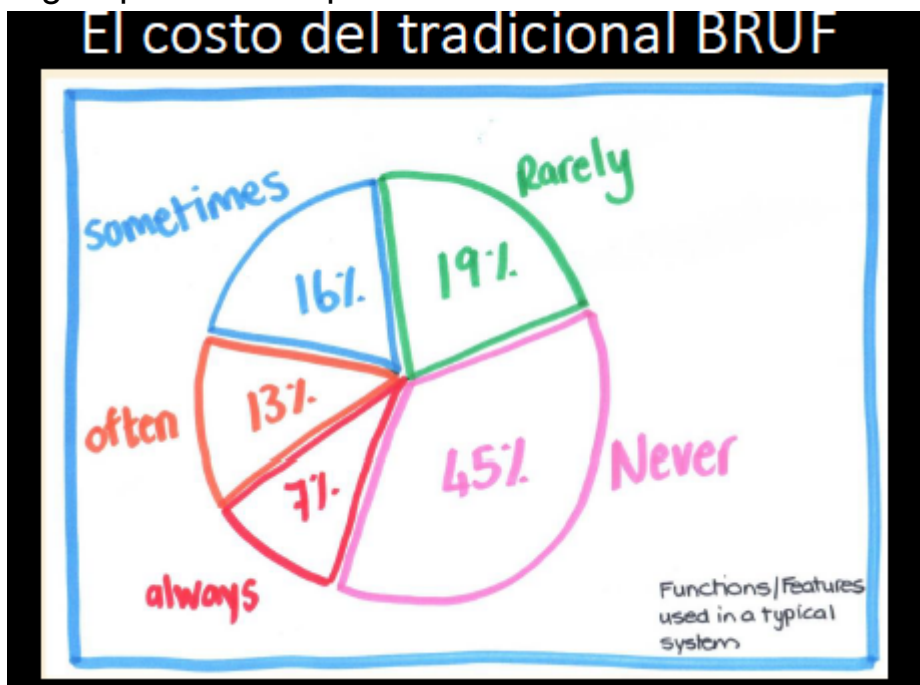
Usamos el "Valor" para construir el producto correcto.

Usamos historias y modelos para mostrar que construir. Con las user stories acordamos las características del producto, son un medio para encontrar las necesidades.

Determinas que es "sólo lo suficiente". Aceptar que los requerimientos van a cambiar, vamos a comenzar trabajar con un producto que no está 100% definido. Partimos de una idea o intención de producto, una visión.

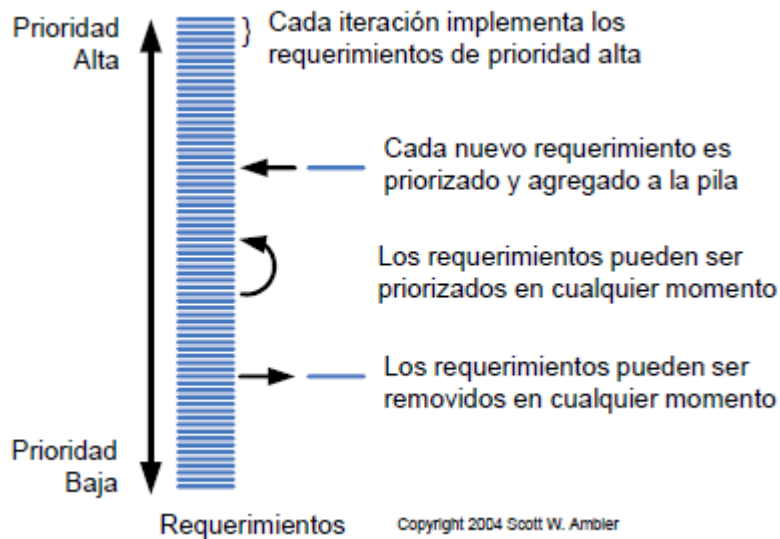
Todo esto se sustenta en el *Costo del tradicional BRUF*

Big requirements up front.



## Product backlog

LA priorización se da desde arriba hacia abajo. Contiene las características del producto, es gestionado por el PO.



## Just in time

Diferimos las decisiones lo más que podamos. Vamos a trabajar con un conjunto acotado de requerimientos, no todos, el que entra en esta iteración. Esto se sustenta en que el 50% de los requerimientos son emergentes. surgen a lo largo del ciclo de vida, no todos se definen desde un instante inicial o BRUF.

Sobre el otro 50% vos tenes conocidos y no cocidos(inexperencia, ineficiencia, mala praxis). El agilismo permite atacar esto.

## Tipos de requerimientos

Siempre que hablamos de requerimientos estamos hablando del dominio del problema y tambien de la solucion. Los CU pueden definir requerimientos del negocio y de usuario. Las US solo son de USUARIO. PERO AMBAS ESTÁN EN EL DOMINIO DEL PROBLEMA.



DUDA PREGUNTAR

## Tradicional vs. Ágil



## Principios Ágiles relacionados a los Requerimientos Ágiles



1- LA PRIORIDAD ES SATISFACER AL CLIENTE A TRAVÉS DE RELEASES TEMPRANOS Y FRECUENTES (2 SEMANAS A UN MES)



2 -RECIBIR CAMBIOS DE REQUERIMIENTOS, AUN EN ETAPAS FINALES



4 - TÉCNICOS Y NO TÉCNICOS TRABAJANDO JUNTOS TODO EL PROYECTO



6 - EL MEDIO DE COMUNICACIÓN POR EXCELENCIA ES CARA A CARA



11 - LAS MEJORES ARQUITECTURAS, DISEÑOS Y REQUERIMIENTOS EMERGEN DE EQUIPOS AUTOORGANIZADOS

## User story

Técnica, es una descripción corta de una funcionalidad esperada por un usuario. Termina siendo un recordatorio de un relevamiento, una historia de un usuario que indica que tenemos que hacer.

Es un token para conversación

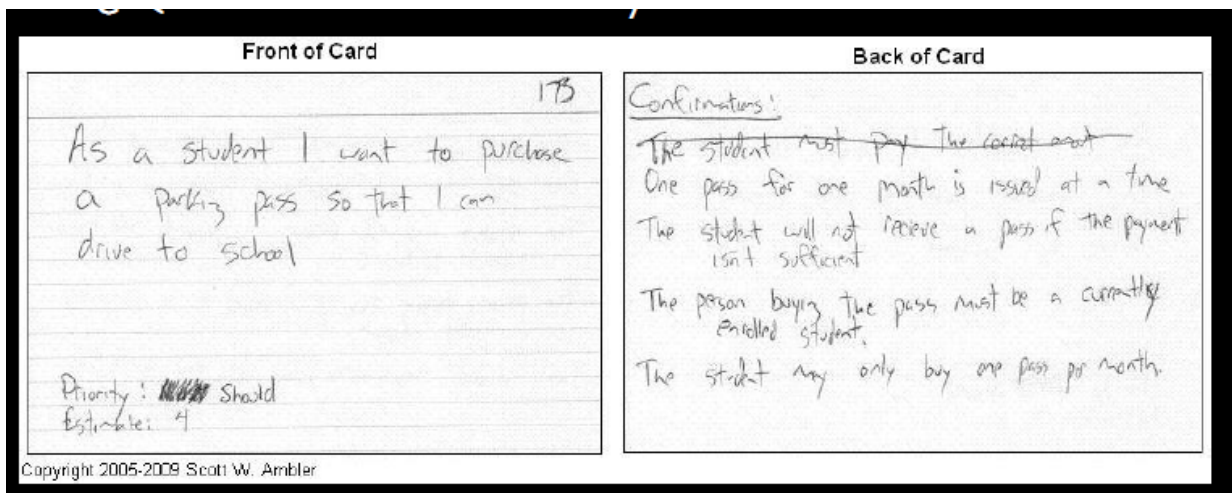
*Leer no silver bullet y the mythical man month.*

## Partes

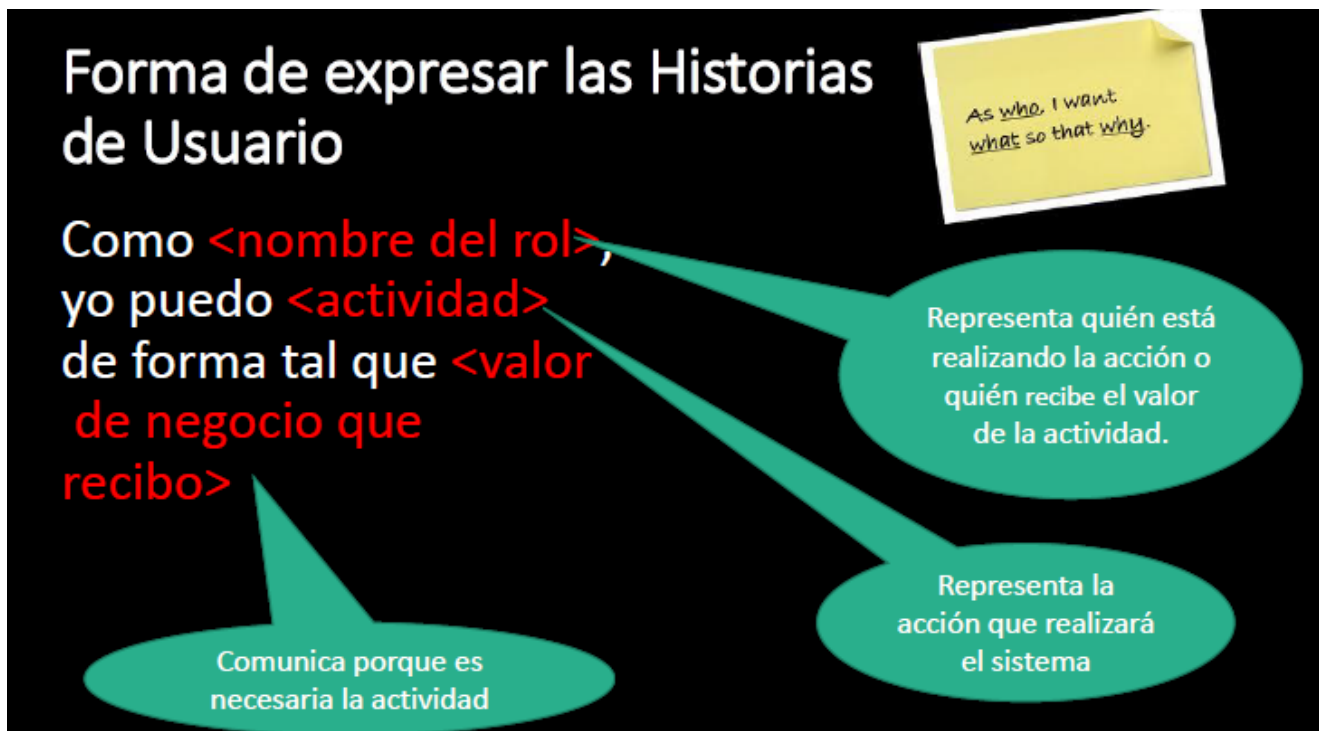
Las tres c:

- card: Es la parte visible. El front de la US
- conversation: Es el elemento más importante. La descripción del pedido por parte del Product Owner.

- Confirmation. Se logra con las pruebas de confirmación. Se usa para chequear si la user story está terminada. Estas pruebas se vinculan o forman a partir de criterios de aceptación



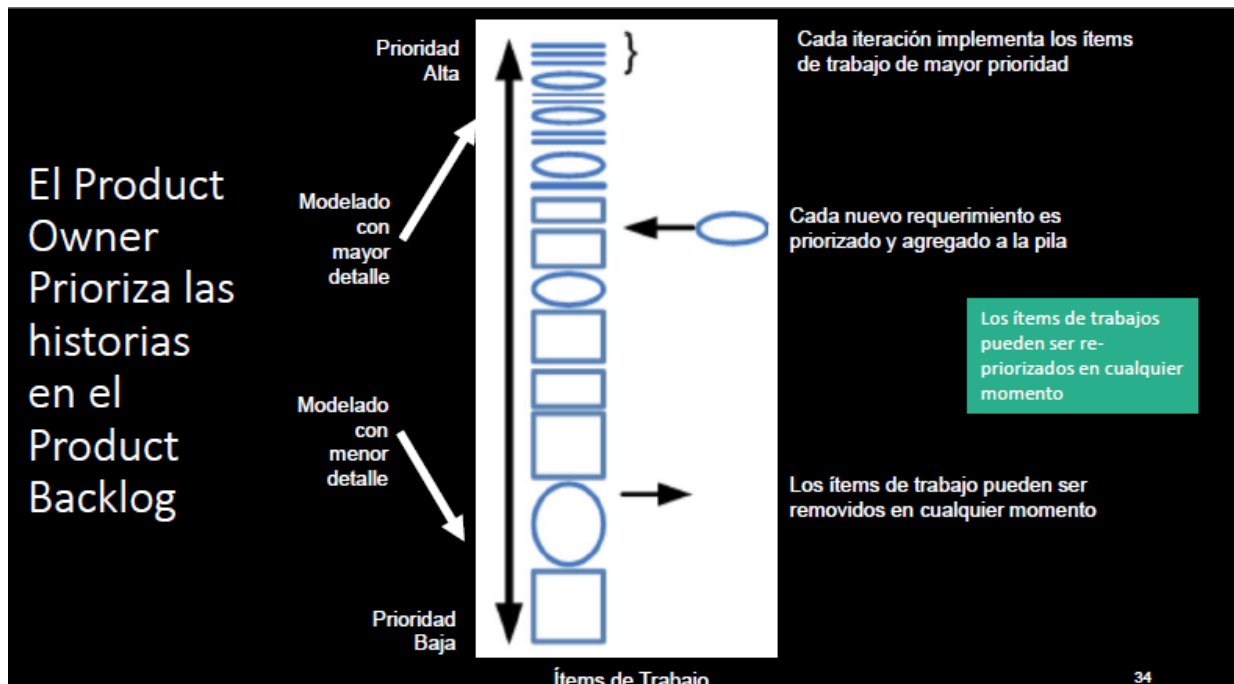
## Sintaxis



- Pensar en para qué lo quiere -> Es el valor de negocio que recibe con esta característica de software.
- La frase verbal se lo puede pensar como una etiqueta identificatoria de la User story, relacionada al qué, se expresa con un verbo en infinitivo.



- Las historias son:
  - Una necesidad del usuario
  - Una descripción del producto
  - Un ítem de planificación
  - Token para una conversación
  - Mecanismo para diferir una conversación



La granularidad es inversamente proporcional a la prioridad.

En un PB puede haber producto en forma de: US's, defectos, deuda técnica, Épicas, organizadores de épicas y US's (Temas).

Es temporal y dinámico.

## Entregables

Lo que se entrega son porciones verticales del producto.

## Criterio de aceptación de user stories:

Ayudan al momento de presentar una US ver si se acepta o rechaza. Definen los límites para una user story. Pueden ser RNF solo cuando el rnf no es transversal o Reglas de negocio.

- Definen límites para una user story (US)
- Ayudan a que los PO respondan lo que necesitan para que la US provea valor (requerimientos funcionales mínimos)
- Ayudan a que el equipo tenga una visión compartida de la US
- Ayudan a desarrolladores y testers a derivar las pruebas.
- Ayudan a los desarrolladores a saber cuando parar de agregar funcionalidad en una US

Incluyendo detalles, con info que ayudan a implementar las us.

Convenciones: **debe** algo, **puede** hacer algo.

- Definen una intención, no una solución
  - Ej.: El usuario debe elegir al menos una cuenta para operar
- Son independientes de la implementación
- Relativamente de alto nivel, no es necesario que se escriba cada detalle

## User Story: un ejemplo de tarjeta

### Buscar Destino por Dirección

Como Conductor quiero buscar un destino a partir de una calle y altura para llegar al lugar deseado sin perderme.

#### Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

Las users story deben hacerse completas:

Pruebas de aceptación: Se derivan de los criterios de aceptación. Como mínimo cada criterio debería tener una prueba de aceptación y una de rechazo. Se escriben como *Probar acción (pasa - falla)*

Expresan detalles resultantes  
de la conversación

## Complementan la User Story

Proceso de dos pasos:

1. Identificarlas al dorso de la US.
2. Diseñar las pruebas completas

### Buscar Destino por Dirección

Como Conductor quiero buscar un destino a partir de una calle y altura para poder llegar al lugar deseado sin perderme.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

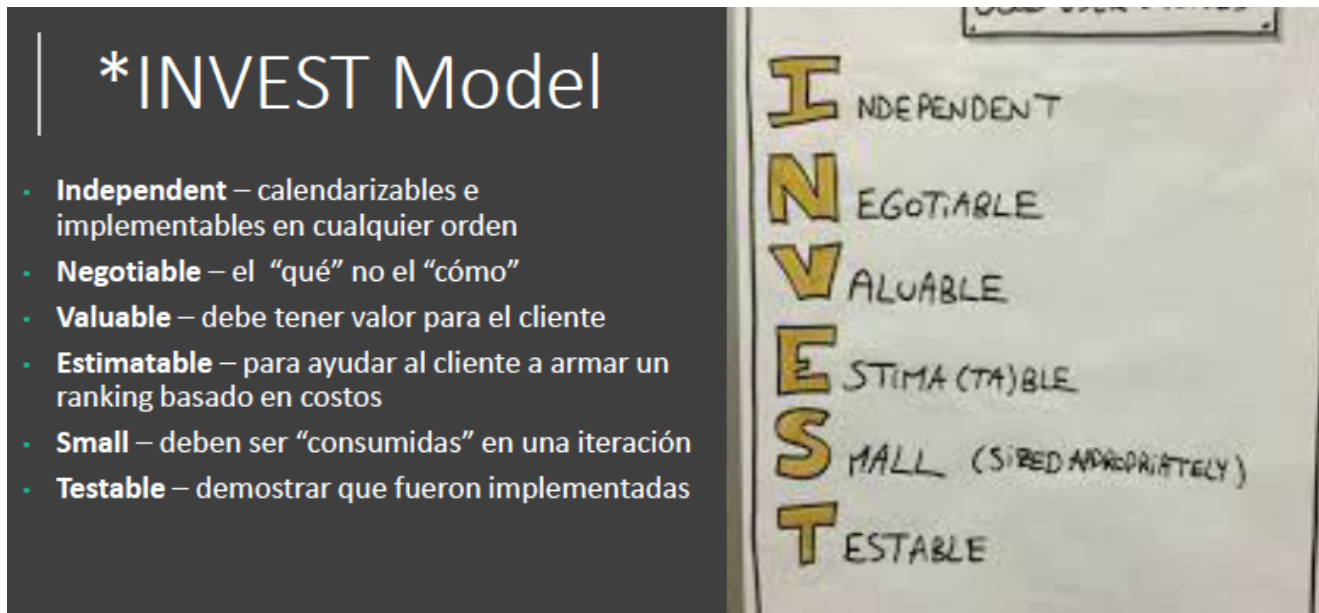
### Pruebas de Usuario

- ☐ Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa).
- ☐ Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla).
- ☐ Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla).
- ☐ Probar buscar un destino en un país inexistente (falla).
- ☐ Probar buscar un destino en País existente, ciudad inexistente (falla).
- ☐ Probar buscar un destino en un país y ciudad existentes, de una calle existente y demora más de 30 segundos (falla).

Varias user storys pueden formar parte de un caso de uso. Las user stories no son validas para documentar.

## INVEST Model

Modelo de autoevaluación que permite evaluar la calidad de las US's



- Independiente: No depende de otras US's para ser implementada. Técnicas: MOCK DE DATOS.
- Negociable: Expresada en términos de qué, no en cómo. Negociar el qué.
- Valuable: Debe tener el para qué, que efectivamente de valor al cliente.
- Estimable: permita darle un valor o ranking o nivel. Si la voy a poder terminar o no en el periodo establecido. Si no se puede estimar, pasa a ser un spike, un elemento de investigación, una user que no se puede implementar, puede ser tanto técnica o funcional, algo del negocio. Termina siendo un item de configuración
- Small: debe ser consumidas en una iteración, deben ser resueltas, por ende desaparecían del Product backlog de la próxima iteración. Que quede implementada al 100%, no acepta trabajos a medias.
- Testable: Demostrar que fueron implementadas. Que podamos probarlas. Saber si cumple o no cumple (que cuente con las pruebas)

## Lean