



**UTN.BA**  
UNIVERSIDAD TECNOLÓGICA NACIONAL  
FACULTAD REGIONAL BUENOS AIRES

**Centro de  
e-Learning**

# **Diplomatura en Metodologías Ágiles**

## **Módulo 2**

### **Lean, KanBan y Scrum**

Derechos Reservados

**Centro de e-Learning SCEU UTN - BA.**

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

**[www.sceu.frba.utn.edu.ar/e-learning](http://www.sceu.frba.utn.edu.ar/e-learning)**



## **Módulo 2 - Unidad 7**

### **Scrum**

Derechos Reservados



## **Presentación:**

El Módulo 2 introduce a los participantes en los Marcos de Trabajo más difundidos dentro de la temática: Lean, Kanban y SCRUM.

Se persigue aquí entender como combinar mejor a estos Marcos de Trabajo entre sí, pudiendo tomar elementos de uno de ellos y emplearlo en otro o también crear modelos nuevos que combinen a los Marcos ya existentes.

Derechos Reservados



## Objetivos:

### Que los participantes logren:

- Identificar los principales Artefactos, Roles y Ceremonias del marco de trabajo de Scrum.
- Comprender los Valores Fundacionales de Scrum.
- Discernir cuando usar Scrum y cuando no.

Derechos Reservados



## Bloques temáticos:

1. Introducción. Cohesión de las Ceremonias y Scrum Orgánico.
2. Roles y Artefactos de Scrum
3. Descripción paso a paso de las ceremonias de Planificación, de Reunión Diaria, de Revisión, de Retrospectiva. Refinamiento
4. Backlog de Producto, Historias de Usuarios, Puntos de Historias, Backlog/Tablero de Sprint
5. Diagrama de Burndown, Incremento de Funcionalidad, Definición de Terminado.
6. Técnicas Avanzadas Refinamiento de Backlog y de Planificación de Entregas. Slicing



## Consignas para el aprendizaje colaborativo

En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC\*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.

*\* El MEC es el modelo de E-learning colaborativo de nuestro Centro.*



## Tomen nota:

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

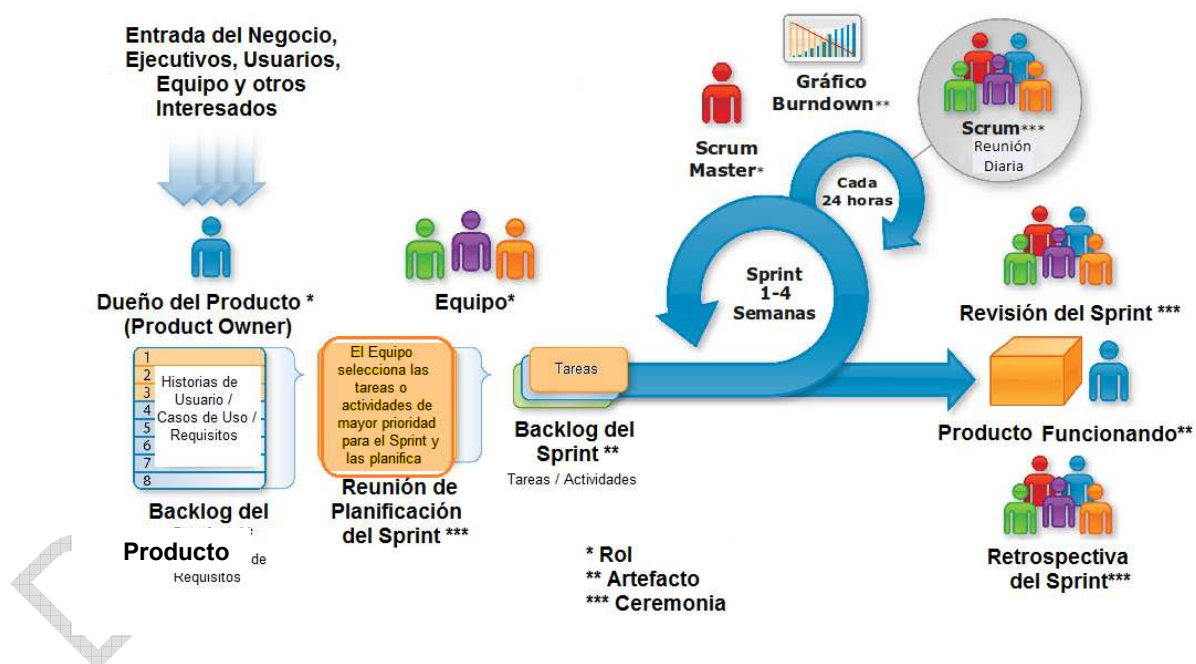
## 1. Introducción. Cohesión de las Ceremonias y Scrum Orgánico.

Scrum es un marco de trabajo ágil, el más utilizado. La mayoría de las veces que las personas dicen que trabajan en ágiles están utilizando el marco Scrum.

Scrum utiliza un proceso iterativo e incremental donde se ejecutan iteraciones de duración corta, manteniendo a un ritmo parejo en el paso por la planificación, la ejecución y la reflexión sobre lo ocurrido.

Cuando Jeff Sutherland creó el proceso de Scrum en el 1993, utilizó el término “scrum” de una analogía presentada en un estudio de 1986, por Takeuchi y Nonaka<sup>1</sup>. En este estudio se comparan equipos de alto rendimiento y multidisciplinario a la formación del “scrum” de los equipos de rugby.

A continuación, se presenta un diagrama y una explicación del ciclo de Scrum:



<sup>1</sup> Hirotaka Takeuchi e Ikujiro Nonaka (1986) - *New New Product Development Game* - Harvard Business Review





- El **Dueño del Producto** crea una lista priorizada de requisitos llamada **Backlog de Producto**.
- Durante la **Planificación** (2-4hs), el **Equipo** selecciona una pequeña porción de esta lista, un **Backlog de Sprint**, y decide cómo implementar estos requisitos.
- El equipo tiene un tiempo limitado, un **Sprint** (1 a 4 semanas), para terminar su trabajo, pero se reúne cada día para constatar su progreso en la **Reunión Diaria** (15 minutos).
- En este camino, el **Scrum Master** mantiene el equipo enfocado hacia su objetivo.
- Al final del sprint, el trabajo debería ser **Potencialmente Entregable**, o sea estar listo para enviar a un cliente, empaquetar para su distribución o presentar a un sponsor.
- El sprint termina con una **Revisión de Sprint** y una **Retrospectiva**.
- Cuando se inicia el próximo sprint, el equipo selecciona otra pequeña porción del backlog de producto y empieza a trabajar de nuevo.

Este ciclo se repite hasta que el backlog de producto se haya completado, que el presupuesto se haya gastado o que un hito definitivo haya llegado, según la especificidad de cada proyecto.

**Sprint** es el nombre de la iteración en Scrum. Se acuerda una duración fija para todos los sprints, en general entre una y cuatro semanas. Los sprints se ejecutan uno tras el otro respetando esta duración, sin tiempo muerto entre el sprint que termina y el sprint que empieza. El objetivo del sprint es de transformar un conjunto acordado de ítems del Backlog de Producto en un Incremento de Funcionalidad de Producto Potencialmente Entregable.

## Cohesión de las Ceremonias

Scrum describe un marco de trabajo esencial (se definen pocos elementos: 3 roles, 4 ceremonias y 3 artefactos) con elementos muy cohesivos entre sí. El trabajo de diseño inicial de Scrum por Jeff Sutherland y Ken Schwaber en el 1993 fue dirigido por un enfoque minimalista centrado únicamente en prácticas de gestión de proyectos y de equipos. El objetivo fue seleccionar buenas prácticas reconocidas y probadas de la industria, y construir un marco mínimo de trabajo a implementar de forma simple, pero con un gran potencial. Su adopción masiva y su éxito en el desarrollo de software y otros dominios demuestran lo acertado de estas definiciones esenciales.

A la hora de implementar Scrum, es importante reconocer la importancia de cada ceremonia en el marco de trabajo, y reflexionar fuertemente a la hora de adoptar parcialmente las ceremonias, para entender lo que se puede perder al no implementar una ceremonia o parte de una ceremonia. Está claro que no siempre se puede o se quiere implementar Scrum en su totalidad,



pero es fundamental ver el marco completo como un objetivo (que puede ser lejano) a perseguir continuamente hacia la agilidad.

Hay duplas de ceremonias que se combinan naturalmente y pierden su efectividad si no se implementan juntas, como por ejemplo **Planificación** y **Revisión**, ya que permiten definir y luego validar el compromiso asumido por un equipo en un sprint, o **Planificación** y **Reunión Diaria**, ya que permiten definir y luego monitorear día a día el avance del equipo.

De la misma forma, sacar una ceremonia del marco de trabajo puede limitar fuertemente el crecimiento ágil del equipo. Por ejemplo, no implementar la ceremonia de **Retrospectiva** restringe drásticamente la mejora continua del equipo, o no implementar la ceremonia de **Reunión Diaria** restringe fuertemente la comunicación del equipo, lo cual suele generar problemas y re-trabajos.

## Scrum Orgánico

Sin embargo, es posible adoptar las ceremonias de Scrum gradualmente, en un enfoque incremental, para adaptarse paso a paso a la cultura o al contexto de la organización considerada. De hecho, es bastante común que los primeros intentos de implementación de Scrum sean incompletos, por ejemplo, sin **Retrospectiva**, o sin **Reunión Diaria**. El enfoque de **Scrum Orgánico**<sup>2</sup> adopta justamente la adopción iterativa e incremental de las ceremonias de Scrum, como un posible camino hacia la agilidad. No hay una receta mágica de implementación de Scrum, y cada implementación tiene su contexto, restricciones, culturas y recursos, por lo cual es aceptable buscar una implementación progresiva. En este proceso progresivo, la experiencia indica que nunca hay que perder de vista los puntos siguientes:

---

<sup>2</sup> Scrum Orgánico por Hiroshi Hiromoto - <http://scrumorganico.com/>



**1**

- Tener claro lo que se deja afuera, lo que se pierde al dejarlo afuera y porqué se lo está dejando afuera.

**2**

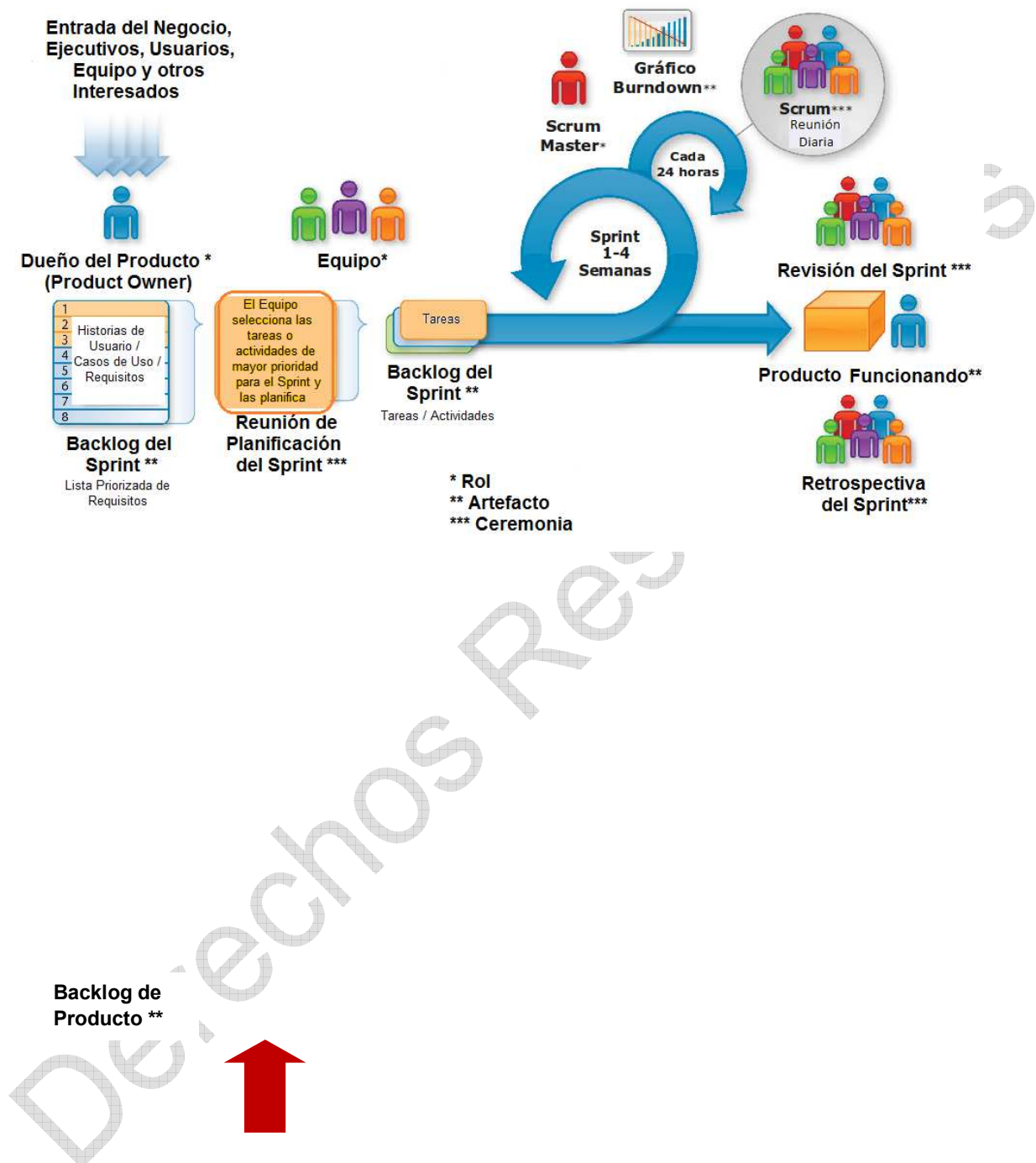
- Apesar de todas las excusas imaginables de antemano, probar seriamente una ceremonia es casi el único medio para poder evaluar realmente su relación costo/beneficio.

**3**

- Periódicamente es importante reflexionar sobre las ceremonias implementadas, para ver si se tienen que adaptar y cuáles otras ceremonias se podrían implementar a continuación.

En lo que queda de la unidad estaremos presentando una a una las principales ceremonias de Scrum y algunas prácticas adicionales comunes a muchos equipos.

Derechos Reservados





## 2. Roles y Artefactos de Scrum

Scrum formaliza tres roles: el *Dueño de Proyecto*, el *Scrum Master* y el *Equipo*.

- **Dueño de Producto**

Es la voz del cliente, el que establece una visión sólida del producto a desarrollar y prioriza continuamente los requisitos para alcanzar dicha visión.

Sus principales responsabilidades:

- Canalizar las necesidades del negocio
- Maximizar el valor para el negocio
- Inspeccionar y adaptar el producto

- **Scrum Master**

El Scrum Master es un líder servicial para el equipo, pero también un facilitador y agente de cambio para la aplicación adecuada de Scrum en el equipo y la organización.

Sus principales responsabilidades:

- Asegurar la correcta aplicación de Scrum.
- Eliminar los impedimentos que frenan el trabajo del equipo.

- **Equipo**

El equipo es un grupo auto-organizado, multidisciplinario y con autonomía, que desarrolla el producto. Su única responsabilidad es de transformar el Backlog de Producto en incrementos de funcionalidad potencialmente entregable en cada sprint. Se caracteriza también el equipo por su auto-gestión.

## **Artefactos**

Scrum identifica tres artefactos o productos de trabajo: el *Backlog de Producto*, el *Backlog de Sprint*, y el *Incremento de Funcionalidad Potencialmente Entregable*.

- **Backlog de Producto**

El Backlog de Producto es una lista única, pública y dinámica que recopila una secuencia priorizada de requerimientos para el producto. Algunos de estos requerimientos tienen una estimación de alto nivel de esfuerzo o complejidad. Se suele decir que el Backlog de Producto representa el “Qué” esperado del producto, sin preocuparse por el “Cómo”.

- **Backlog de Sprint**

El Backlog de Sprint es un conjunto reducido negociado de ítems del Backlog de Producto que el equipo se compromete a completar durante el tiempo alocado al Sprint. Cada ítem incluido en el sprint es dividido en tareas detalladas que el equipo va a completar, con una estimación de duración que no supere un día de esfuerzo. El Backlog de Sprint es actualizado diariamente por el equipo, y muestra en cada momento las tareas pendientes, en curso y terminadas para completar los ítems del sprint, así como una estimación del esfuerzo pendiente de cada tarea no terminada y a quién está asignada la tarea. Se suele utilizar un *Tablero de Sprint* como forma de visualizar el Backlog de Sprint, o algún otro soporte herramental que permite darle visibilidad.

- **Incremento de Funcionalidad Potencialmente Entregable**

Al final de cada sprint el equipo debería entregar un incremento de funcionalidades del producto con una calidad productiva. El incremento del producto debe funcionar y si quedan tareas pendientes para su entrega, deberían requerir un esfuerzo mínimo.



### **3. Descripción paso a paso de las ceremonias de Planificación, de Reunión Diaria, de Revisión, de Retrospectiva. Refinamiento**

Scrum identifica cuatro ceremonias o prácticas a ejecutar en cada sprint: la *Planificación*, la *Reunión Diaria*, la *Revisión* y la *Retrospectiva*.

- **Planificación**

Objetivos:

- ☐ Entender y determinar el trabajo que el Equipo se compromete a terminar en el Sprint.
- ☐ Definir cómo se va a realizar este trabajo.

- **Reunión Diaria**

Objetivos:

- ☐ Compartir en el equipo el avance del trabajo comprometido para el Sprint.
- ☐ Coordinar las actividades del equipo para terminar el trabajo comprometido para el Sprint.

- **Revisión**

Objetivos:

- ☐ Demostrar concretamente y claramente el progreso del equipo.
- ☐ Recibir retroalimentación (feedback) periódica de los usuarios/clientes sobre los productos/resultados generados.

- **Retrospectiva**





#### Objetivos:

- ☐ Escuchar distintos puntos de vista dentro del equipo sobre lo ocurrido en el Sprint.
- ☐ Identificar colaborativamente las causas de los principales problemas del equipo durante el Sprint.
- ☐ Idear, consensuar y seleccionar acciones de mejora concretas que pueda ejecutar el equipo en el próximo Sprint.

Todo backlog de producto suele empezar con una etapa de descubrimiento de producto, o también llamada *Backlog Grooming*, *Backlog Refinement*, *Pre-Planning* o *Pre-Planificación*, el **Refinamiento de Backlog** tiene como objetivo mejorar y preparar el **Backlog de Producto** para las próximas ceremonias de **Planificación**.

Suele ocurrir una o varias veces durante el **Sprint**, en general unos días antes de la próxima **Planificación**, según el estado del **Backlog de Producto** y las necesidades del **Dueño de Producto**<sup>3</sup>.

*Backlog Grooming*, *Backlog Refinement*, *Pre-Planning* o *Pre-Planificación*, el **Refinamiento de Backlog** tiene como objetivo mejorar y preparar el **Backlog de Producto** para las próximas ceremonias de **Planificación**.

Es una práctica muy importante que permite minimizar el tiempo de la **Planificación** y mejorar su calidad.

#### El Backlog de Producto **DEEP**<sup>4</sup>

Se usa la sigla **DEEP** (profundo) para definir cómo debería ser un buen **Backlog de Producto**:

- **D**etallado apropiadamente. Los elementos de mayor prioridad están más detallados y especificados que los de menor prioridad, ya que se trabajará antes en los primeros que en

<sup>3</sup> Es recomendable hacer entre una o dos sesiones de máximo 4 horas de duración por de Refinamiento de **Backlog** por **Sprint**.

<sup>4</sup> Adaptado del *Scrum Primer*, por Pete Deemer, Gabrielle Benefield, Craig Larman y Bass Vodde - <http://scrumprimer.org>





los últimos. Por ejemplo, el 10% inicial del **Backlog de Producto** puede estar compuesto elementos muy pequeños y bien analizados, mientras que el 90% restante pueden estarlo mucho menos.

- **E**stimado. Los elementos del **Backlog de Producto** que componen la próxima versión del producto (*release* o entrega) deben estar estimados y deberían revisarse y re-estimarse en cada **Sprint** a medida que se disponga de mayor información/contexto.
- **E**mergente. Debido a la variabilidad y al aprendizaje experimentado, el **Backlog de Producto** es refinado regularmente. En cada **Sprint** se pueden añadir, eliminar, modificar, dividir o cambiar la prioridad de los elementos. Así, el **Dueño de Producto** actualiza constantemente el **Backlog de Producto** para reflejar cambios en las necesidades del cliente, nuevas ideas o percepciones, movimientos de la competencia, cambios regulatorios, etc.
- **P**riorizado. Los elementos del **Backlog de Producto** están priorizados de manera secuencial, de 1 a N. En general, los elementos de máxima prioridad deberían crear el mayor valor por el dinero: deberían producir el máximo valor (de negocio) a cambio de poco dinero (coste). Otro posible motivo para priorizar un elemento es atacar los riesgos de forma temprana.

## Actividades del Refinamiento de Backlog

Para lograr un **Backlog de Producto DEEP**, el **Equipo** y el **Dueño de Producto** trabajan juntos (con la facilitación del **Scrum Master**), ejecutando una a varias de las actividades siguientes:



**Depuración de elementos**

**Ordenamiento de elementos**

**Análisis de requisitos**

**División de elementos grandes en varios elementos más pequeños**

**Unión de varios elementos**

**Estimación o re-estimación de elementos**

Derechos Reservados



## 4. Backlog de Producto, Historias de Usuarios, Puntos de Historias, Backlog/Tablero de Sprint

El artefacto sobre el cual se basa la ceremonia de **Planificación** es el **Backlog de Producto**.

### **BACKLOG DE PRODUCTO**

*El **Backlog de Producto** es una lista ordenada de ideas para el producto, mantenidas en el orden en el cual se espera que se resuelvan. Es la fuente única desde la cual fluyen todos los requerimientos. Eso implica que todo el trabajo del equipo viene del Backlog de Producto. Por ejemplo, cada idea de funcionalidad, mejora, resolución de incidente, necesidad de documentación se deriva de un ítem del Backlog de Producto. Cada ítem del **Backlog de Producto** incluye una descripción y una estimación de alto nivel.*

*El **Backlog de Producto** puede empezar como una larga o corta lista, puede ser de muy alto nivel o muy detallado. Generalmente empieza como una lista corta y de muy alto nivel, y con el paso del tiempo se vuelve más larga y detallada. Los elementos que van a ser implementados próximamente suelen ser “refinados”: clarificados, mejor definidos, divididos en varios elementos más chicos, como parte de la actividad de **Refinamiento de Backlog de Producto**.*

*El **Dueño de Producto** es responsable de mantener el **Backlog de Producto**, aunque puede – y debería – ser ayudado para producir y actualizarlo. Los elementos del **Backlog de Producto** pueden ser originados por el **Dueño del Producto**, por los miembros del equipo o por cualquier otro involucrado.*

Traducido y adaptado del Agile Atlas

<http://agileatlas.org>

A continuación, se muestra un ejemplo de un **Backlog de Producto** en formato de planilla:



Prioridad	Elemento	Estimación	Estado	Criterios de Aceptación
1	Como usuario registrado, quiero poder comprar acciones para administrarlas más adelante	3	En curso	<ul style="list-style-type: none"><li>● Probar una compra simple</li><li>● Probar con acciones de tipo A, B y C</li><li>● Probar con usuario sin saldo</li><li>● Probar con usuario no registrado</li></ul>
2	Como usuario no registrado, quiero poder buscar eventos de mercado para ver si me interesan algunos	5	Terminado	<ul style="list-style-type: none"><li>● Probar con rango de fecha valido e invalido</li><li>● Probar con múltiples eventos un mismo día</li></ul>
3	Como usuario registrado, quiero recibir información detallada de una empresa elegida para estudiar su perfil financiero	3	Pendiente	<ul style="list-style-type: none"><li>● Probar con empresas sin ficha detallada</li><li>● Probar con empresa con ficha detallada</li><li>● Probar con usuario registrado</li></ul>
4	Como administrador de inversiones, quiero obtener un aviso de alertas para anticipar posibles fraudes	8	Pendiente	<ul style="list-style-type: none"><li>● Probar por email, RSS y SMS</li><li>● Probar con múltiples emails en un solo campo</li><li>● Probar selección de un tipo de alerta o de todos a la vez</li></ul>



Se suele utilizar el formato de **Historias de Usuario** (*User Story*) como medio para recoger los requerimientos de producto de manera ágil y registrarlos en el **Backlog de Producto**.

### **DEFINICIÓN**

Una **Historia de Usuario** describe una funcionalidad que será de valor para un usuario o cliente. Las historias de usuarios se componen de tres aspectos:

- Una descripción escrita de la historia empleada para planificar y como recordatorio
- Conversaciones sobre la historia que sirven para canalizar sus detalles
- Pruebas que sostienen y documentan detalles y que pueden ser empleadas para evaluar si la historia fue terminada

Mike Cohn

Se suele emplear el siguiente formato para documentar la historia de usuario:

***Como <rol>, puedo <funcionalidad>, para <beneficio de negocio>***

Por ejemplo: “Como usuario registrado del sitio de turismo online, quiero reservar nuevos viajes para que se actualicé mi bonus.”

Otro ejemplo: “Como cliente de MovieOnline, quiero buscar películas por actor, para que pueda encontrar más fácilmente las películas que quiero alquilar.”

Emplear este formato para el análisis ágil obliga al equipo a identificar claramente los usuarios de las funcionalidades a construir, y también a describir al producto desde la perspectiva del negocio y de los beneficios a lograr.

Es bastante común garantizar la efectividad de las historias de usuarios con el mecanismo **INVEST**.



Cuando más actualizado y definido este el **Backlog de Producto** para la ceremonia de **Planificación**, mejor va a ser el resultado de la misma. Este refinamiento suele hacerse en reuniones de **Refinamiento de Backlog** (ver la sección 6.1- *Refinamiento de Backlog*).

- **Contar con el espacio adecuado**

El horario de la ceremonia de **Planificación** debe ser acordado con anticipación. Como todos los **Sprints** tienen la misma duración, es muy común acordar un horario fijo de la **Planificación** para todos los **Sprints**<sup>5</sup>.

En la **Parte I** de la **Planificación**, deben participar el **Dueño de Producto**, el **Scrum Master** y el **Equipo**. En la **Parte II**, deben participar el **Scrum Master** y el **Equipo**, y es recomendable que el **Dueño de Producto** siga disponible (quizás por teléfono). Es necesario que todos los participantes estén presentes en los horarios acordados para las dos partes de la **Planificación**, en el lugar acordado.

## Paso a Paso (Parte I)

La **Parte I de la Planificación** se centra en “*qué*” es lo que el **Dueño de Producto** quiere que se desarrolle en el **Sprint** y “*por qué*” es necesario.

Si bien existen distintas formas de conducir la **Parte I de la Planificación**, destacamos la siguiente, de las más adoptadas:

### 1. Definición de Sprint

El **Scrum Master** agenda o repasa con el **Equipo** y el **Dueño de Producto** las fechas claves del **Sprint**:

---

<sup>5</sup> Por ejemplo “el primer día del sprint de 9h a 11h”.



- Inicio y fin del **Sprint**, y eventuales feriados o días que no se pueden contar en el **Sprint**.
- Fecha, horario y lugar de las ceremonias de **Reunión Diaria**, **Revisión** y **Retrospectiva**.

## 2. Revisión de Elementos

El **Dueño de Producto** y el **Equipo** revisan los elementos de mayor prioridad del **Backlog de Producto** que al **Dueño de Producto** le interesa desarrollar durante este Sprint. Se discuten los objetivos, la motivación y el contexto correspondiente a cada uno de los elementos, de tal forma que el **Dueño de Producto** pueda transmitir al **Equipo** la visión del producto a desarrollar, y se aclaran todas las dudas que puedan surgir. Si no estuviera hecho anteriormente, se pueden definir criterios de aceptación para cada uno de los elementos.

## 3. Definición del Objetivo del Sprint

El **Dueño de Producto** y el **Equipo** pueden formular un objetivo del **Sprint**. Se trata de una declaración que resume la finalidad del **Sprint**, en el caso que este mismo tenga un tema central<sup>6</sup>.

El objetivo del **Sprint** proporciona al **Equipo** flexibilidad sobre el alcance comprometido, ya que, aunque puedan tener que eliminar algunos elementos en caso de no llegar a terminarlos antes del final del **Sprint**, tendrían que comprometerse a entregar algo tangible y “terminado” que esté alineado con el espíritu del objetivo del sprint.

Este paso es opcional.

## 4. Estimaciones de Alto-Nivel

---

<sup>6</sup> Por ejemplo: “Desarrollo de las funcionalidades de *Leasing*”, “Puesta a punto de los problemas de Base de Datos”, etc.





El **Equipo** determina una estimación de alto nivel de cada uno de los elementos del **Backlog de Producto** presentados, en el caso que no lo haya estimado previamente (por ejemplo, en **Sprints** anteriores o en reuniones de **Refinamiento de Backlog**). Para eso puede ser necesario tomar algunas decisiones iniciales de diseño y hacer preguntas adicionales al **Dueño de Producto** para asegurar un buen entendimiento del ítem.

Una característica bastante común de las estimaciones en las metodologías ágiles es que sean **relativas**. Eso significa que no se busca una estimación universal de la complejidad o del esfuerzo de construcción un elemento del **Backlog de Producto**, sino una comparación de su complejidad contra otros elementos conocidos. En particular no se busca una estimación en horas del esfuerzo de construcción del elemento, sino poder comparar su complejidad de desarrollo contra la de otros elementos.

Para lograr este objetivo, se suele usar una unidad de estimación abstracta de comparación entre elementos, como por ejemplo el **Punto de Historia** (*Story Point*).

#### **DEFINICIÓN**

Un **Punto de Historia** es una unidad arbitraria pero fija que describe cuánto esfuerzo requiere un elemento del **Backlog de Producto** para ser entregado al cliente.

Esta medida relativa suele ser útil para pensar en forma más abstracta las estimaciones y compararlas contra elementos conocidos. La escala de *Puntos de Historia* y su correspondencia promedia con el esfuerzo son características propias de cada equipo de proyecto.

Se suelen usar técnicas de estimaciones como *Planning Poker*<sup>7</sup>, *T-Shirt Sizing*<sup>8</sup>, *Wideband Delphi*<sup>9</sup> u otras<sup>10</sup>.

<sup>7</sup> Agile Estimating (Planning Poker) <http://www.mountaingoatsoftware.com/presentations/114-agile-estimating>

<sup>8</sup> Estimating with Tee Shirt Sizes <https://www.mountaingoatsoftware.com/blog/estimating-with-tee-shirt-sizes>  
o T-Shirt Sizing to Estimate Development Effort in a Software Project  
<http://www.brighthub.com/office/project-management/articles/22847.aspx>

<sup>9</sup> Wideband Delphi <http://www.slideshare.net/aniruddha.chakrabarti/wideband-delphi-estimation>

<sup>10</sup> Affinity Estimating <https://scrumology.com/affinity-estimating/>





## 5. Acuerdo Preliminar de Alcance

A partir de las prioridades expuestas por el **Dueño de Producto** en el paso 2. Y de las estimaciones definidas en el paso 4., el **Equipo** determina el alcance del **Sprint**: que elementos del **Backlog de Producto** se comprometen a terminar.

Esta determinación es responsabilidad del **Equipo**, y nadie más lo puede definir. Para determinar cuántos elementos comprometer, el Equipo se basa en su performance en Sprints anteriores, también llamada *Velocidad del Equipo* o en la mejor proyección de capacidad del **Equipo** posible.

Una vez definido este alcance preliminar del **Sprint**, se valida y eventualmente se ajusta con el **Dueño de Producto**.

## Paso a Paso (Parte II)

La **Parte II de la Planificación** se centra en “cómo” implementar los elementos que el **Equipo** decide incluir en el Sprint.

Si bien existen distintas formas de conducir la **Parte II de la Planificación**, destacamos la siguiente, de las más adoptadas:

### 1. Identificación de tareas

El **Equipo** colabora para identificar todas las tareas necesarias para terminar cada uno de los elementos del **Backlog de Producto** comprometidos para el **Sprint**, de acuerdo al **Criterio de Terminado** acordado (ver sección 4.2).

---

o Affinity Estimating – A How-To <http://scrumology.com/guest-post-affinity-estimating-a-how-to/>  
o Swimlane Sizing – Complete & Fast Backlog Estimation <http://theagilepirate.net/archives/109>  
o ¿Cansado del Planning Poker? 4 alternativas <http://www.elproximopaso.net/2012/06/cansado-del-planning-poker-4.html>



Se busca identificar para cada elemento del **Backlog de Producto** todas las tareas necesarias para llegar al incremento de funcionalidad potencialmente entregable correspondiente (ver sección 4.2). Por ejemplo, se suelen identificar tareas de relevamiento adicional, diseño de base de datos, diseño de interfaz gráfica, programación, armado de lotes de pruebas, pruebas, etc.

En general se suele trabajar con una granularidad de tareas de entre 1h y 8h. En caso de tareas más cortas, se busca combinar varias tareas en una más grande, y en caso de tareas más largas, se busca dividir las en sub-tareas.

## 2. Estimaciones de tareas

El **Equipo** estima el esfuerzo requerido para cada una de las tareas identificadas. Se registra el esfuerzo en horas de cada tarea, y si fuera necesario se agrupa o se divide la tarea para llegar a la granularidad esperada.

Se busca una estimación de esfuerzo promedio del **Equipo**, sin asignar a la persona quien va a ejecutar la tarea.

## 3. Chequeo de Alcance

El **Scrum Master** ayuda al **Equipo** a realizar distintos chequeos para validar la factibilidad de cumplir con el alcance preliminar comprometido para el **Sprint** (por ejemplo, horas estimadas vs. horas disponibles del equipo).

Si fuera necesario se acuerda un cambio al alcance del Sprint con el **Dueño de Producto**.

De esta forma queda fijado el alcance del **Sprint**.

## 4. Armado Backlog de Sprint / Tablero de Sprint



En este paso, el **Scrum Master** ayuda al **Equipo** a terminar de registrar o armar el **Backlog de Sprint / Tablero de Sprint**.

### **BACKLOG DE SPRINT**

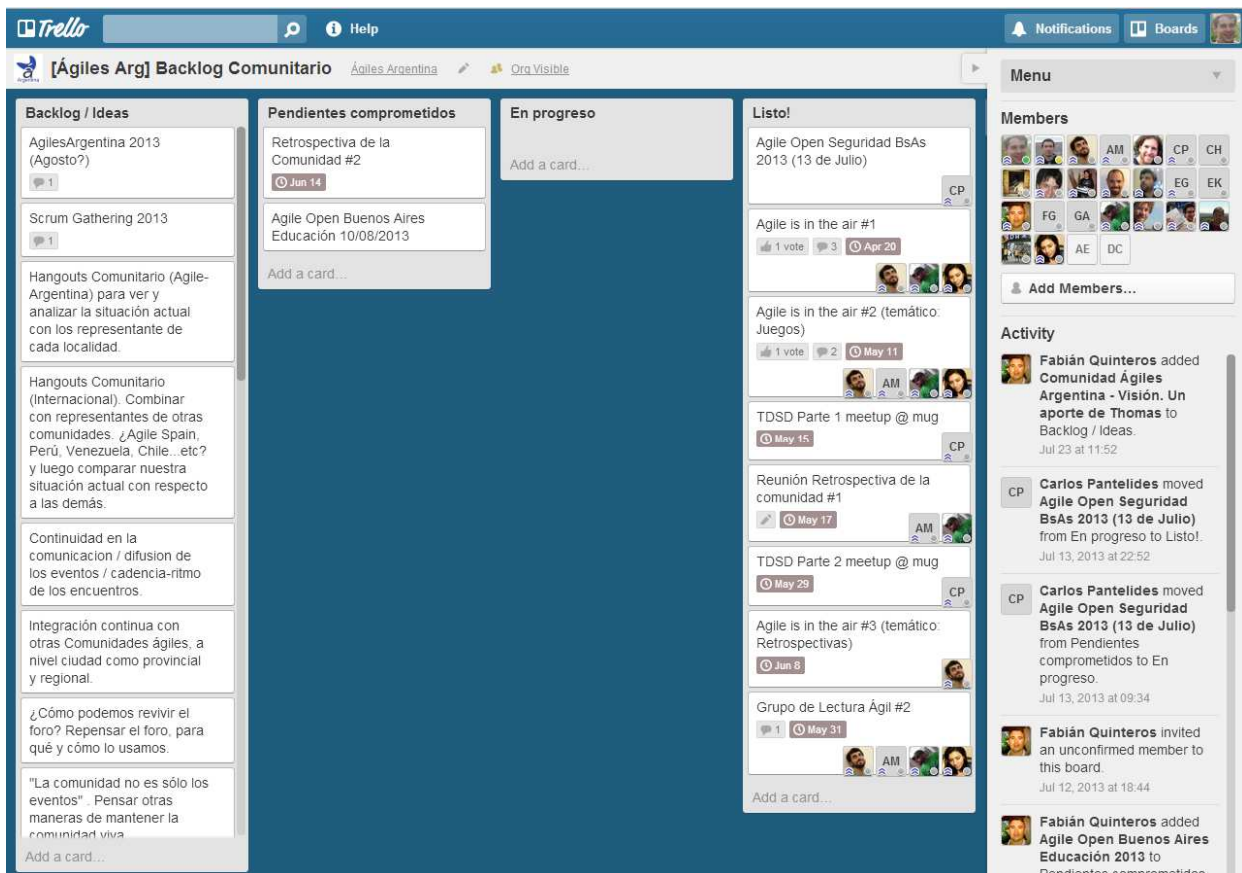
El **Backlog de Sprint** es un conjunto reducido negociado de elementos del **Backlog de Producto** que el **Equipo** se comprometió a completar durante el **Sprint**. Cada elemento incluido es dividido en tareas detalladas que el **Equipo** va a completar, con una estimación de duración que no supere un día de esfuerzo. El **Backlog de Sprint** es actualizado diariamente por el equipo, y muestra en cada momento las tareas pendientes, en curso y terminadas para completar los elementos comprometidos, así como una estimación del esfuerzo pendiente de cada tarea no terminada y a quién está asignada la tarea.

Se suele utilizar un **Tablero de Sprint** como forma de visualizar el **Backlog de Sprint**, o algún otro soporte herramental digital que permite darle visibilidad.

A continuación, se ve un ejemplo de **Backlog de Sprint** con la herramienta *Trello*<sup>11</sup>:

---

<sup>11</sup> <https://trello.com/>



Ejemplo de Backlog de Sprint

### **TABLERO DE SPRINT**

Una práctica común en Scrum cuando todo el **Equipo** de desarrollo está colocado en un mismo lugar es la de materializar el **Backlog de Sprint** a través de un tablero físico (por ejemplo, con rotafolios, panel de corcho, pizarrón, etc.) expuesto en la sala donde trabaja el **Equipo**. De esta forma se logra una visibilidad importante sobre el avance del **Sprint** para todas las personas que están ubicados en este lugar o que pasan por la sala.



El **Tablero de Sprint** presenta el **Backlog de Sprint** visualmente, usualmente con papeles pegados (post-its) o tarjetas de cartón o papel para representar ítems y tareas del **Sprint**, en las cuales se registra información básica de los elementos comprometidos (como por ejemplo título, estimación de esfuerzo, prioridad, criterios de aceptación) y de las tareas (como por ejemplo tarea, esfuerzo restante estimado, responsable). Se suele dividir el tablero en tres columnas, con los tres estados de tareas: Pendiente, En Curso, Terminado, en las cuales se ubican las tareas según el avance.

A continuación, se ve un ejemplo de **Tablero de Sprint**:



*Ejemplo de Tablero de Sprint*

Se prepara también el **Diagrama de Burndown** correspondiente al **Sprint** (ver sección 3.3)



## 5. Asignación de Primeras Tareas

El **Equipo** determina en conjunto las primeras tareas que se van a realizar en las próximas 24h (o hasta la próxima **Reunión Diaria**), y se asignan las responsabilidades correspondientes.

Derechos Reservados





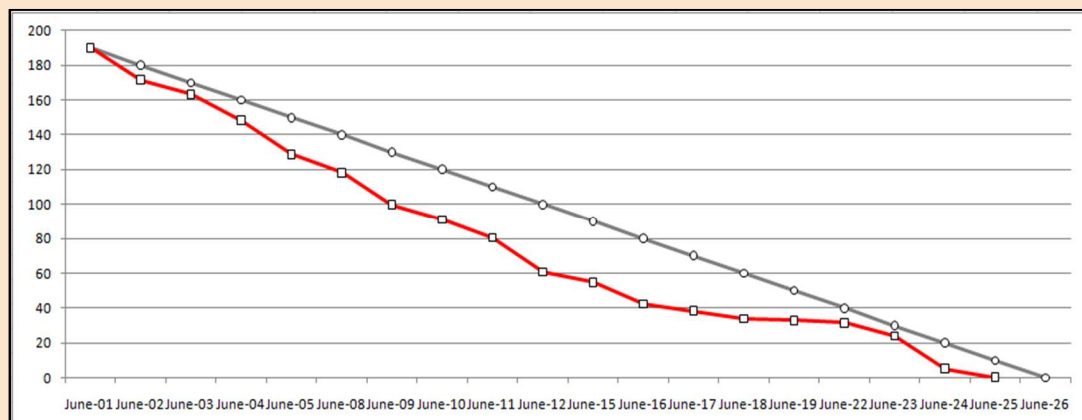
## 5. Diagrama de Burndown, Incremento de Funcionalidad, Definición de Terminado.

El **Scrum Master** se asegura que este actualizada la información del **Tablero de Sprint** o **Sprint Backlog**.

En particular se actualiza el **Diagrama de Burndown**.

### DIAGRAMA DE BURNDOWN

El **Diagrama de Burndown** es un componente del **Tablero de Sprint** o **Backlog de Sprint** muy importante para el monitoreo del avance del **Sprint**.



**Ejemplo de Burn-Down Chart**

Este gráfico muestra, cada día, una estimación de cuánto trabajo queda hasta que el **Equipo** haya terminado. Idealmente, se trata de un gráfico descendente que se va aproximando a “cero esfuerzo pendiente” el último día del **Sprint**. Por eso se llama un diagrama de burndown (ir “quemando” trabajo hacia abajo). Lo importante es que muestra el progreso hacia el objetivo, no en términos de cuánto tiempo se ha invertido en el pasado, sino en términos de cuánto trabajo queda en el futuro. Si la línea del burndown no va cuesta abajo hacia la finalización del trabajo cerca del final del **Sprint**, el **Equipo** necesita realizar ajustes, como por ejemplo reducir el alcance del trabajo o encontrar maneras de trabajar más efectivamente manteniendo un ritmo sostenible.



*En este ejemplo vemos un sprint de 20 días de duración, con un esfuerzo restante estimado inicial de 190 horas. En gris se ve la línea de progreso teórico y en rojo el registro diario del esfuerzo restante actualizado.*

*El Diagrama de Burndown permite visualizar distintas situaciones:*

*- Si la pendiente sube, se puede ver que el esfuerzo restante está creciendo. Puede pasar cuando se están agregando tareas no previstas, o que surgen problemas que hacen que las tareas duren más de lo estimado.*

*- Si la pendiente es horizontal de un día al otro, quiere decir que el esfuerzo restante no cambia de un día al otro, o sea que los eventuales avances fueron compensados por incrementos en esfuerzos restantes estimados (nuevas tareas, complicaciones en tareas existentes, etc.).*

*- Si la pendiente pasa por arriba de la línea de progreso teórico, se puede interpretar como que el equipo se esté atrasando y que probablemente no se pueda cumplir con los objetivos del sprint siguiendo este ritmo.*

*- Si la pendiente pasa por debajo de la línea de progreso teórico, se puede interpretar que el equipo está avanzado más rápido que lo previsto (puede ser que se eliminaron algunas tareas, que algunas tareas se terminaron en menos tiempo que lo previsto, etc.).*

*Tener este gráfico actualizado diariamente y muy visible (o accesible vía una herramienta) es un mecanismo muy potente de monitoreo. Poder ver muy simplemente el esfuerzo restante para cumplir con los objetivos del **Sprint** es un indicador de avance muy expresivo, y cualquier persona relacionada con el proyecto puede darse cuenta muy rápidamente si la iteración va bien o si ocurren algunos problemas a resolver.*

El **Scrum Master** confronta el avance actual contra lo comprometido del **Sprint**. El **Equipo** evalúa cómo llegar al objetivo comprometido, y eventualmente se re-organizan las tareas pendientes en consecuencia.





Durante el **Sprint**, el **Equipo** trabajó en los elementos comprometidos del **Backlog de Producto** para generar un **Incremento de Funcionalidad**.

#### **INCREMENTO DE FUNCIONALIDAD**

*En cada **Sprint** el **Equipo** desarrolla un **Incremento de Funcionalidad** del producto. Debe tener la calidad adecuada para ser potencialmente entregable o utilizable por los usuarios/clientes representados por el **Dueño de Producto**. También se deben minimizar las tareas pendientes para entregarlo o ponerlo en uso<sup>12</sup>.*

*Durante el **Sprint**, el **Equipo** se auto-organiza para generar el **Incremento de Funcionalidad** del producto de acuerdo al **Backlog de Sprint**, como comprometido durante la **Planificación**.*

Los elementos del **Backlog de Producto** que componen el **Incremento de Funcionalidad** son desarrollados de acuerdo a los estándares de calidad de la organización y a la **Definición de Terminado** adoptada por el **Equipo**.

#### **DEFINICIÓN DE TERMINADO (Done)**

*Cuando se entrega el **Incremento de Funcionalidad**, debe estar “terminado” de acuerdo a un entendimiento común de lo que “terminado” significa. Esta definición es distinta para cada equipo de Scrum, y va a crecer y ser más drástica a medida que el equipo madure.*

*La **Definición de Terminado** siempre debe considerar la noción de que el **Incremento de Funcionalidad** tiene una calidad suficiente para ser entregado: el **Dueño de Producto** podría decidir su entrega inmediata.*

---

<sup>12</sup> Por ejemplo, las tareas de pruebas, de integración, pasajes de ambientes, documentación de manuales, aceptación de usuarios, *builds*, etc.



*El **Incremento de Funcionalidad** incluye la funcionalidad de todos los **Incrementos de Funcionalidad** de **Sprints** anteriores y debe ser completamente probado para asegurar que todos los elementos del **Backlog de Producto** terminados siguen funcionando juntos.*

Traducido y adaptado del Agile Atlas

<http://agileatlas.org>

En consecuencia, el **Equipo** debe preparar la ceremonia de Revisión para poder mostrar el **Incremento de Funcionalidad “Terminado”** funcionando. Puede ser preparar esta demostración en un ambiente lo más parecido al ambiente productivo. Ciertos equipos definen un guión de la revisión que determina en qué orden mostrar las funcionalidades, con qué datos, por quién, de qué forma, etc. Algunos equipos ensayan la revisión antes de la ceremonia.

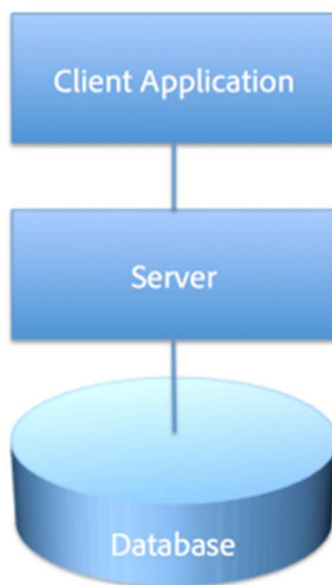
En caso de no haber “terminado” todos los elementos del Backlog de Producto comprometidos, el **Equipo** determina cómo se va a mostrar (o no) el avance en cuanto a los elementos no “terminados”.

Derechos Reservados



## 6. Técnicas Avanzadas Refinamiento de Backlog y de Planificación de Entregas. Slicing

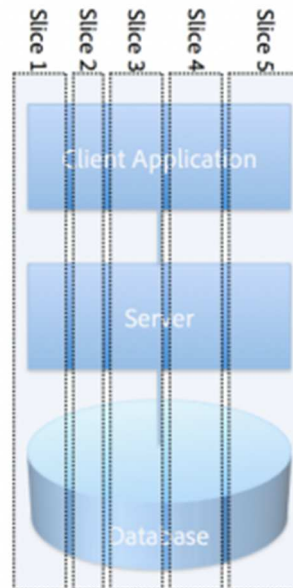
Cuando tratamos de llevar el paradigma de trabajo tradicional a la división de historias nos puede llevar a hacerlo utilizando los llamados **cortes horizontales** en los cuales hacemos los cortes guiados por las capas de arquitectura; siendo por ejemplo, realizar las modificaciones en la base de datos, la capa lógica y por último la interfaz usuario, luego de lo cual se integra y se prueba el funcionamiento integral.



### ¿Por qué cortar las historias?

Esta organización permite una alta especialización de cada área busca en teoría asegurar organización, y si bien puede sonar intuitivo y lógico tiene como puntos negativos la demora en obtener feedback y el trabajo en “silos” que atenta contra mirada más holística del paradigma ágil.

En cuando buscamos acelerar el feedback necesitamos buscar cortes que atraviesan diversas capas y nos permitan tener software operativo buscando que los diversos roles realicen sus partes de manera altamente integrada y colaborativa para su integración, testeo y rápida posibilidad de release



Este tipo de recorte es el buscado cuando seguimos el paradigma ágil, en inicio puede verse como ineficiente ya que implica varias veces trabajar sobre una misma parte, pero esta supuesta ineficiencia se ve compensada por tener feedback frecuente del usuario para detectar diferencias y ganar aprendizaje sobre software construido.

Existen varias heurísticas para el llamado slicing de historias, detallaremos dos patrones de los autores más reconocidos sobre el tema.

### **I. Los patrones definidos por Richard Lawrence:**

(fuente: <http://agileforall.com/vertical-slices-and-scale/> )

#### ***Patrón 1: Pasos de workflows***

*COMO administrador de contenido QUIERO publicar artículos en el sitio corporativo PARA el acceso de nuestros clientes*

Esta historia puede parecer bastante simple y concreta, hasta que empezamos a indagar en un refinamiento y vemos que de acuerdo al tipo de historia, rol o incluso en todas, este es un proceso de varios pasos que requiere revisiones, correcciones, etc. Las cuales si buscamos realizarlas todas



desde el inicio hacen difícil que se termine la historia en una iteración o al menos que podamos incluir otras.

Una manera que nos reduce la complejidad sería particionar la historia por los distintos pasos que puede tener para construir la opción más simple y directa que podamos mostrar y validar en funcionamiento, sobre la cual luego incorporamos los distintos pasos en otras historias:

Dividiríamos inicialmente en:

*COMO administrador de contenido QUIERO publicar artículos directamente en el sitio corporativo sin validaciones*

Y luego:

*COMO administrador de contenido QUIERO publicar artículos con revisión del editor.*

*... QUIERO publicar artículos con revisión del sector legal.*

*... QUIERO publicar artículos en el sitio de homologación.*

*... QUIERO publicar artículos desde homologación en producción.*

Esta división permite por lo pronto poder publicar directamente en el sitio, que suena lo de mayor valor, quitando las validaciones por parte de las otras áreas que tal vez, sólo suceden en la minoría de los casos o que podamos hacerlo manualmente mientras ya utilizamos el software producido. Justamente por esta mención es que hacer un corte por cada paso sería incorrecto, porque al construir historias de esa manera estaríamos haciendo algo similar a los cortes horizontales por arquitectura, ya que al finalizar cada historia no tendríamos algo operativo.

## ***Patrón 2: Variaciones en las reglas de negocio***

Este tipo de patrón puede darse cuando existen distintas reglas complejas para hacer la operación. Así por ejemplo en la siguiente historia:

*COMO usuario QUIERO buscar por mis operaciones bancarias con fechas flexibles*

Al refinar "Fechas flexibles" podemos encontrar diversas reglas:

*COMO usuario QUIERO buscar por mis operaciones bancarias entre X e Y*

*... QUIERO buscar por mis operaciones bancarias en una semana de diciembre*



*... QUIERO buscar por mis operaciones bancarias desde X hasta el día de hoy.*

*... QUIERO buscar por mis operaciones bancarias N días desde X*

Siendo cada una independiente esta división nos permite reducir la complejidad en partes.

### **Patrón 3 Mayor Esfuerzo**

En ocasiones una historia puede contener el mayor esfuerzo en una parte que luego es aprovechada por otras, así por ejemplo puedo tener:

*COMO usuario QUIERO pagar la compra con VISA, MasterCard y American Express.*

En esta historia la mayor complejidad podría estar en la infraestructura para el pago electrónico, luego de preparado para una sería aprovechada para implementar las otras 2.

Por lo cual podríamos escribirlo de la siguiente manera:

*COMO usuario QUIERO poder para con una tarjeta de crédito (de Visa, MasterCard o American Express)*

*... QUIERO poder pagar con los 3 tipos de tarjeta de crédito (de Visa, MasterCard o American Express).*

El no expresar una tarjeta puntual de manera inicial nos da libertad para que el PO priorice a último momento cual es la tarjeta con la cual empezar y de acuerdo a eso, estimar las restantes.

### **Patrón 4 Complejo/Simple**

Es posible que en ciertas historias al ir pensando los distintos escenarios, existen cada vez más opciones y complejidades

En ese sentido, nos conviene priorizar un escenario que nos sirva como la implementación desde la cual agregaremos los siguientes.

Así por ejemplo, podemos tener:

*COMO usuario QUIERO poder realizar la búsqueda de los vuelos entre 2 destinos*



Siendo que existen muchas opciones de búsqueda para esos parámetros podemos dividirlos de la siguiente manera:

*COMO usuario QUIERO buscar vuelos directos entre 2 destinos*

*... QUIERO buscar vuelos con N paradas entre 2 destinos*

*... QUIERO buscar vuelos entre 2 destinos incluyendo aeropuertos cercanos.*

#### **Patrón 5 Variaciones en Datos**

En ocasiones, los tipos de datos que podemos tener son extremadamente variados, lo cual impide cerrar un alcance claro. Esto puede verse por ejemplo, en los sistemas de venta de pasajes, para los cuales las diversas empresas tienen sus propias firmas y reglas internas que permiten distintas opciones de acuerdo a como se lo requiera, en este caso nos conviene seleccionar la que consideramos de más valor (costo/beneficio) e ir agregando opciones a continuación.

*COMO usuario QUIERO buscar pasajes entre 2 destinos*

En este caso podemos iniciar por el proveedor de pasajes que utiliza el formato que nos resulta más práctico y creamos las siguientes historias:

*COMO usuario QUIERO buscar pasajes entre A y B con proveedores que permitan cargar origen y destino indicados como ciudades*

*... QUIERO buscar pasajes entre A y B con proveedores que devuelvan las ciudades más cercanas de cada uno*

#### **Patrón 6 Manera de entrada de datos**

En ocasiones la complejidad puede estar dada por la interfaz usuario en lugar de por la funcionalidad en sí misma. En cuyo caso podemos dividir la historia en la implementación de la funcionalidad en sí misma y luego incorporar las mejoras de UX

*COMO usuario QUIERO poder realizar la búsqueda de operaciones entre 2 fechas (cargando manualmente en formato DD/MM/YYYY)*

*QUIERO poder realizar la búsqueda de operaciones entre 2 fechas con un calendario gráfico.*

#### **Patrón 7 Diferir performance**





En ocasiones el optimizar los tiempos de respuesta pueden requerir un esfuerzo importante para llegar a lo deseado, pero es algo que tal vez podemos diferir para un siguiente paso y poder avanzar en obtener feedback del resto de la funcionalidad (primero hacer que ande, luego hacerlo rápido)

*COMO usuario QUIERO poder buscar las operaciones entre 2 fechas (aunque tenga baja performance, indicar un icono animado de en proceso)  
... QUIERO poder buscar en menos de 5 segundos,*

### **Patrón 8 Operaciones**

En muchos casos se utiliza la palabra "administrar", la cual puede tener aparejada múltiples operaciones que pueden implementarse por separado (y algunas diferirse con algún proceso manual momentáneamente)

Por ejemplo:

*COMO usuario QUIERO administrar mi cuenta*

Puede dividirse en:

*COMO usuario QUIERO poder crear una cuenta*

*... QUIERO editar los datos de mi cuenta*

*... QUIERO eliminar mi cuenta.*

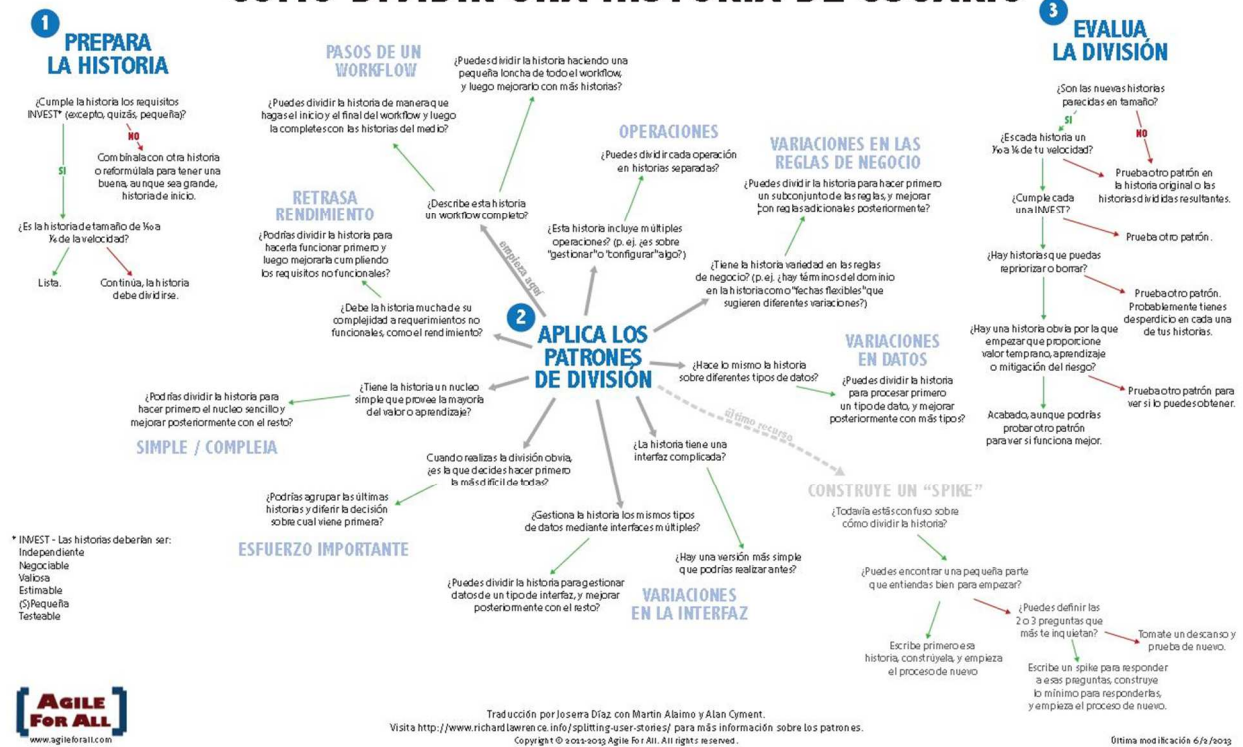
### **Patrón 9 Crear una Spike**

Una historia puede ser costosa a causa de una alta incertidumbre, en cuyo caso, lo aconsejable es hacer una spike con duración fija que nos permita obtener información sobre la cual decidir. Debemos ser claros en los objetivos a cubrir y los mismos no deberían ser ambiguos o demasiado extensos.

Por último, incluyo el diagrama realizado por el autor para guiarnos en esto. Puede resultarnos muy útil para recordarlo durante el refinamiento.



## COMO DIVIDIR UNA HISTORIA DE USUARIO



### S.P.I.D.R de Mike Cohn

Mike Cohn creó un enfoque para desagregar historias de usuario rápidamente. Él lo denomina S.P.I.D.R. (práctica común en los estadounidenses de usar anagramas, en este caso similar a Spider- araña en inglés-). Esta heurística es un tanto más simple que el anterior, aunque tiene algunas prácticas compartidas. Los patrones son:

- **Spike:** En lugar de tener una historia grande con alto nivel de incertidumbre, mejor hacer una pequeña investigación desde la cual obtener los datos requeridos para que el equipo obtenga más información. Con esta información poder estimar correctamente y facilitar la desagregación de historias en tamaños manejables
- **Paths (Camino o secuencia):** En estos casos podemos evaluar los flujos incluidos en la historia y dividir cada uno en una historia distinta. También se puede considerar que cada paso será una historia en sí misma.



- **Interfaces:** Una opción bastante usada, que suele ser simple, es dividir las historias por interfaces, incluso si sigue siendo demasiado grande por tecnología (explorador o dispositivo). Es muy recomendable iniciar con una construcción mínima de la interface dejando su estilización para un paso posterior.
- **Data (Datos):** En muchos casos podemos necesitar trabajar con múltiples formatos de datos. Considerar cuáles son los más usados o críticos, puede ser una buena manera para dividir la historia, incluso utilizar una versión simplificada para probar alguna hipótesis, puede probar ser útil.
- **Rules (Reglas):** En ocasiones, la existencia de validaciones, tecnologías, estándares, etc., pueden complejizar las historias. Reducir su aplicación en primeras versiones es una práctica útil para dividir las historias, incorporando las reglas en historias posteriores.

### Prácticas usuales

A las prácticas recomendadas por estos reconocidos autores, agrego varias que considero útiles:

- Por tipos de usuario, de acuerdo a las necesidades de los diversos roles y “personas”, evaluar sus prioridades y valor económico para saber por dónde empezar.
- Diferir performance y arquitectura en general. En este caso es particularmente útil al inicio de la vida del producto, donde queremos validar la hipótesis de negocio antes de invertir más recursos.
- Diferir la interfaz de usuario, al menos las funcionalidades no críticas o puramente estéticas. Este enfoque obviamente es más simple en backoffice que en frontends de productos de empresas .COM.
- Según los datos, podemos inicialmente no aceptar todos los diversos tipos o formatos de datos haciendo foco en el/los más comunes. Así por ejemplo, sólo trabajamos con medidas en metros/ kilómetros y luego incorporamos la posibilidad de trabajar en millas y pulgadas.
- Crear una Spike cuando tenemos mucha incertidumbre, haciendo foco en aprender.
- El camino "feliz" completo sacrificando toda validación e ir incorporando progresivamente nuevos caminos o validaciones.
- Por Reglas de negocio. Este caso similar al anterior, uno puede demorar su implementación o realizarlas manualmente.
- “Mockear” las interacciones que incluyan a sistemas externos. Esto nos permite destrabar el avance y poder validar lo demás hasta tener la interfaz formal.



"Paretizar" las funcionalidades, para quedarnos con lo más valioso hasta llegar a un tamaño "manejable".

Derechos Reservados



## Bibliografía utilizada y sugerida

- <https://www.scrumguides.org/>. (2019). Retrieved from <https://www.scrumguides.org/>.
- Institute, P. M. (2017). *A Guide to the Project Management Body of Knowledge - PMBOK 6th*. Project Management Institute.
- Institute, P. M. (2017). *Agile Practice Guide*. Project Management Institute.
- Institute, P. M. (2018). *Organizational Project Management Maturity Model*. Project Management Institute.
- Nir, M. (2017). *The Agile PMO*. Amazon Digital Services.



## Lo que vimos:

- SCRUM



## Lo que viene:

- Retrospectivas

