



Una introducción básica a la teoría y práctica de Scrum
Versión 2.0

Pete Deemer
GoodAgile
www.goodagile.com

Gabrielle Benefield
Evolve
www.evolvebeyond.com

Craig Larman
www.craiglarman.com

Bas Vodde
Odd-e
www.odd-e.com



Nota a los lectores: hay muchas descripciones concisas de Scrum disponibles on-line, y esta introducción pretende proporcionar el siguiente nivel de detalle sobre sus prácticas. No debe considerarse como el paso definitivo en el aprendizaje de Scrum: es aconsejable que los equipos que estén considerando adoptar Scrum adquieran *Agile Project Management with Scrum* de Ken Schwaber, *Succeeding with Agile* de Mike Cohn y aprovechen cualquiera de las muchas excelentes opciones de formación y coaching existentes (detalles completos en scrumalliance.org). Nuestro agradecimiento a Ken Schwaber, Dr. Jeff Sutherland, y Mike Cohn por sus generosas sugerencias.

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Traducción al castellano:

Ángel Medinilla

Proyectalis

www.proyectalis.com/AngelMedinilla

Más allá del desarrollo tradicional

El desarrollo tradicional basado en grupos especializados, ciclos de feedback débiles o tardíos, planificación predictiva “a priori” y flujo secuencial desde el análisis hasta las pruebas no está teniendo mucho éxito en el cambiante mundo de hoy en día. Esta estrategia de desarrollo retarda el feedback, el aprendizaje y el potencial retorno de inversión debido a una ausencia de software funcionando hasta las fases finales del proyecto, lo que provoca falta de transparencia, escasez de posibilidades para mejorar, reducción de la flexibilidad y un incremento de los riesgos técnicos y de negocio.

Durante décadas ha existido una alternativa – equipos cross-funcionales realizando un desarrollo iterativo – pero no ha sido tan empleada como el modelo tradicional. Scrum engloba conceptos simples y comprobados para el desarrollo de productos, en un *Framework* (marco de trabajo) simple que incluye auténticos equipos, equipos cross-funcionales, equipos auto-gestionados, ciclos completos de feedback cortos e iterativos y reducción del coste de los cambios. Estos conceptos incrementan la agilidad y el feedback, permiten un ROI más temprano y reducen el riesgo.

Visión general

Scrum es un marco de trabajo en el que equipos cross-funcionales pueden crear productos o desarrollar proyectos de una forma iterativa e incremental. El desarrollo se estructura en ciclos de trabajo llamados Sprints (también conocidos como iteraciones). Estas iteraciones no deben durar más de cuatro semanas cada una (siendo dos semanas la duración más habitual) y tienen lugar una tras otra sin pausa entre ellas. Los Sprints están *acotados en el tiempo* – finalizan en una fecha determinada independientemente de si el trabajo ha finalizado por completo o no, y *jamás se prorrogan*. Normalmente los equipos Scrum escogen una duración de Sprint y la mantienen para todos sus Sprints hasta que mejoran y pueden emplear ciclos más cortos. Al principio de cada Sprint, un **Equipo cross-funcional** (de en torno a siete personas) selecciona **elementos** (peticiones del cliente) de una lista priorizada. El equipo acuerda un objetivo colectivo respecto a lo que creen que podrán entregar al final del Sprint, algo que sea tangible y que estará “terminado” por completo. Durante el Sprint no se podrán añadir nuevos elementos; Scrum se adapta a los cambios en el siguiente Sprint, pero el pequeño Sprint actual está pensado para concentrarnos en un objetivo pequeño, claro y relativamente estable. Todos los días el Equipo se reúne brevemente para inspeccionar su progreso y ajustar los siguientes pasos necesarios para completar el trabajo pendiente. Al final del Sprint, el Equipo revisa el Sprint con los diferentes Stakeholders (interesados e involucrados en el producto) y realiza una demostración de lo que han desarrollado. Se obtiene feedback que podrá ser incorporado en el siguiente Sprint. Scrum enfatiza un producto “funcionando” al final del Sprint que esté realmente “terminado”. En el caso del software, esto significa un sistema que está integrado, testado, con la documentación de usuario generada y potencialmente entregable. Los principales roles, artefactos y eventos están resumidos en Figura 1.

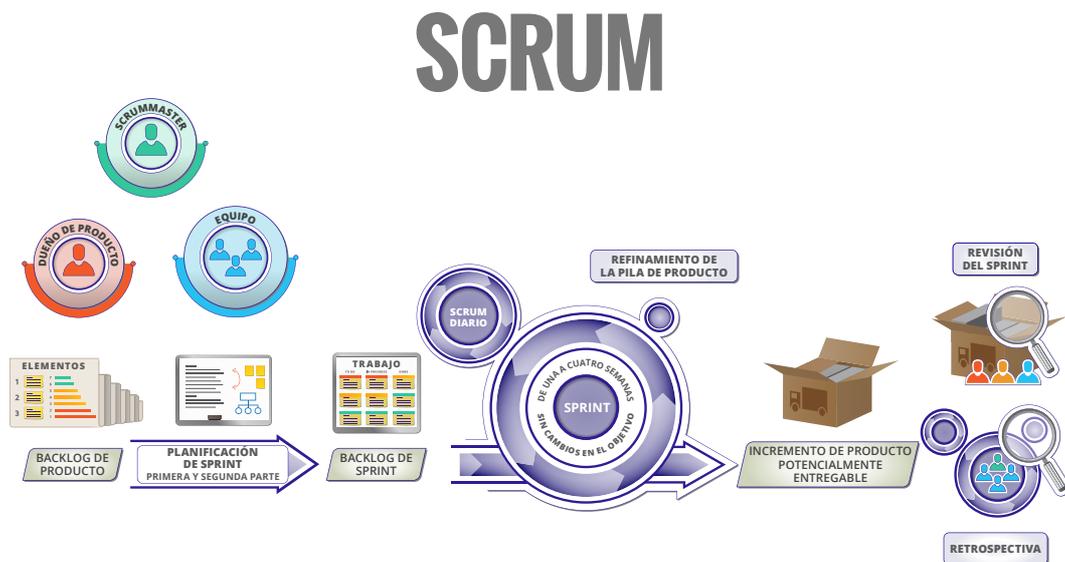


Figura 1. Visión General de Scrum

Un lema recurrente en Scrum es “inspección y adaptación”. Dado que el desarrollo conlleva de forma inevitable aprendizaje, innovación y sorpresas, Scrum enfatiza dar pequeños pasos en el desarrollo, inspeccionando tanto el producto resultante como la eficacia de las prácticas actuales, y a continuación adaptar los objetivos respecto al producto y las prácticas de los procesos. *Repetir indefinidamente.*

Roles en Scrum

En Scrum existen tres roles: Dueño de Product, Equipo y ScrumMaster. Todos ellos forman lo que se conoce como el Equipo Scrum.

El **Dueño de Producto** es responsable de maximizar el retorno de inversión (ROI) a base de identificar las funcionalidades de producto, trasladarlas a una lista priorizada, decidir cuáles deberían estar al principio de la lista para el siguiente Sprint, y repriorizar y refinar continuamente dicha lista. El Dueño de Producto es responsable a nivel ganancias y pérdidas del producto, asumiendo que se trata de un producto comercial. En el caso de aplicaciones internas, el Dueño de Producto no es responsable del ROI en el mismo sentido que en un producto comercial (que generaría ingresos), pero aun sería responsable de maximizar el ROI en el sentido de escoger – en cada Sprint – los elementos que más valor aportan. En la práctica, “valor” es un término muy difuso, y la priorización puede verse influenciada por el deseo de satisfacer a los clientes clave, la alineación con los objetivos estratégicos, atacar riesgos, mejorar y otros factores. En algunos casos el Dueño de Producto y el cliente son la misma persona; esto es frecuente en el caso de desarrollos internos. En otros, “el cliente” pueden ser millones de personas con necesidades muy variadas, en cuyo caso en muchas organizaciones el rol del Product Owner es similar al de Product Manager (Director de Producto) o Product Marketing Manager (Director de Marketing de Producto). Sin embargo, el rol del Dueño de Producto es algo distinto del Product Manager tradicional, ya que interactúa de forma activa y regular con el Equipo, prioriza trabajando con todos los stakeholders y revisa los resultados de cada Sprint, en lugar de delegar las decisiones de desarrollo a un Jefe de Proyecto. Es importante hacer notar que, en Scrum, hay una y sólo una persona que sirve como Dueño de Producto y ejerce la autoridad final como tal, y él o ella es responsable del valor del trabajo realizado, aunque dicha persona no tiene por qué trabajar sola.

El **Equipo** (también llamado **Equipo de Desarrollo**) construye lo que el Dueño de Producto indica: por ejemplo, una aplicación o un sitio Web. El Equipo en Scrum es “cross-funcional” – engloba toda la experiencia y conocimiento necesarios para desarrollar un producto potencialmente entregable en cada Sprint – y es “auto-organizado” (auto-gestionado), con un amplio margen de autonomía y responsabilidad. El Equipo decide cuántos elementos (del conjunto que ofrece el Dueño de Producto) va a desarrollar durante el Sprint y cuál es la mejor manera de lograr dicho objetivo.

Cada miembro del Equipo es simplemente un *miembro de equipo*. Nótese que no hay títulos fijos especializados en un grupo que adopta Scrum: no hay analistas de negocio, administradores de bases de datos, arquitectos, líderes de equipo, diseñadores / especialistas en UX, programadores... Los miembros del Equipo trabajan juntos durante cada Sprint de cualquier manera que sea apropiada para lograr el objetivo que se han marcado ellos mismos.

Dado que solo hay *miembros de equipo*, el Equipo no es sólo cross-funcional, sino que además muestra *aprendizaje múltiple*: cada persona indudablemente tendrá ciertas fortalezas, pero también continuará aprendiendo otras especialidades. Cada persona tendrá habilidades principales, secundarias e incluso terciarias, y se espera de ellos que “vayan a por el trabajo”: que emprendan tareas con las que se sienten menos familiarizados con el objetivo de ayudar a completar un elemento. Por ejemplo, una persona cuya habilidad principal es el diseño de interacción de usuario (UX) podría tener habilidades secundarias en automatización de pruebas; una persona con habilidad principal en redacción técnica podría también ayudar con el análisis y la programación.

El Equipo Scrum tiene siete más/menos dos personas, y en el caso del Software el Equipo podría incluir a personas con habilidades en análisis, desarrollo, pruebas, diseño de interfaz, diseño de bases de datos, arquitectura, documentación y demás. El Equipo desarrolla el producto y proporciona ideas al Dueño de Producto sobre cómo hacer que el producto sea un éxito. En Scrum, los Equipos son más productivos y efectivos si todos los miembros están dedicados al 100% a trabajar en un producto durante el Sprint. El Equipo evita la multi-tarea sobre varios productos o proyectos para evitar el gran coste que tienen la pérdida de concentración y el cambio de contexto (*context switching*). Los Equipos estables se asocian con mayor productividad, por lo que hay que evitar cambiar miembros del Equipo. Las áreas de producto con muchas personas se organizan mediante múltiples Equipos, cada uno de ellos concentrados en diferentes funcionalidades del producto y con alta coordinación de sus esfuerzos. Dado que cada equipo

realiza todo el trabajo necesario para una funcionalidad desde la perspectiva del cliente (planificación, análisis, desarrollo y pruebas), los Equipos se denominan también *feature teams* (equipos de funcionalidad).

El ScrumMaster ayuda al área de producto a aprender y aplicar Scrum para obtener valor de negocio. El ScrumMaster hace todo lo que esté en su mano para ayudar al Equipo, al Dueño de Producto y a la organización a tener éxito. El ScrumMaster *no* es el jefe de los miembros del Equipo, como tampoco es un jefe de proyecto, líder de equipo o representante del equipo. En su lugar, el ScrumMaster *sirve* al Equipo; ayuda a eliminar impedimentos, protege al Equipo de interferencias externas y les ayuda a adoptar prácticas de desarrollo modernas. El o ella forma, entrena y guía al Dueño de Producto, al Equipo y al resto de la organización en el uso correcto de Scrum. El ScrumMaster es un *coach* y un *formador*. El ScrumMaster se asegura de que todo el mundo (incluidos el Dueño de Producto y los gerentes) comprenden los principios y las prácticas de Scrum, y ayudan a guiar a la organización a través de los habitualmente difíciles cambios que son necesarios para lograr el éxito en el desarrollo Ágil. Como Scrum hace visibles múltiples impedimentos y amenazas a la efectividad del Equipo y del Dueño de Producto, es importante contar con un ScrumMaster comprometido trabajando de forma enérgica en ayudar a resolverlos: si no, será difícil que el Equipo o el Dueño de Producto tengan éxito.

El de ScrumMaster debería ser un rol dedicado al 100%, aunque un equipo pequeño podría tener un miembro del equipo ejerciendo dicho Rol (y con una menor carga de trabajo regular en dicho caso). Los mejores ScrumMasters surgen de todo tipo de experiencias y disciplinas: Ingeniería, Diseño, Pruebas, Dirección de Producto, Gestión de Proyectos o Gestión de la Calidad.

El ScrumMaster y el Dueño de Producto no pueden ser la misma persona, ya que su enfoque es muy diferente y combinar ambos roles normalmente conduce a confusión y conflictos. Uno de los desafortunados resultados de combinar ambos roles es un Dueño de Producto haciendo micro-gestión, lo cual es justo lo opuesto al equipo auto-gestionado que requiere Scrum. Al contrario de los gerentes tradicionales, el ScrumMaster no le dice a la gente lo que tiene que hacer o asigna tareas – facilitan el proceso dando soporte al Equipo mientras que este se organiza y se gestiona por su cuenta. Si el ScrumMaster se encontraba anteriormente en una posición de gerencia sobre el equipo, tendrá que realizar un cambio profundo en su forma de pensar y estilo de interacción con el Equipo para tener éxito con Scrum.

Nota: no existe ningún tipo de rol de jefe de proyecto en Scrum. Esto se debe a que no es necesario en absoluto: las responsabilidades tradicionales del jefe de proyecto han sido divididas y reasignadas entre los tres roles de Scrum, sobre todo entre el Equipo y el Dueño de Producto más que en el ScrumMaster. La práctica de Scrum con jefes de proyecto añadidos demuestra un desconocimiento fundamental de Scrum, y típicamente desemboca en conflictos de responsabilidad, confusión respecto a la autoridad y resultados por debajo de lo esperado. A veces, un (ex) jefe de proyecto puede tomar el rol de ScrumMaster, pero el éxito de esta iniciativa depende fuertemente de la persona y de hasta qué punto comprende bien las diferencias fundamentales entre ambos puestos, tanto en las responsabilidades del día a día como en el set mental necesario para tener éxito. Una buena forma de comprender el rol de ScrumMaster a conciencia y comenzar a desarrollar las habilidades clave necesarias para dicho éxito son los cursos de “Certified ScrumMaster” de la Scrum Alliance.

Además de estos tres roles, hay otros stakeholders que contribuyen al éxito del producto, incluyendo los gerentes, clientes y usuarios finales. Algunos stakeholders como por ejemplo los gerentes funcionales (como un director de ingeniería) pueden descubrir que su puesto, aunque sigue aportando valor, cambia al adoptar Scrum:

- Apoyan al Equipo a base de respetar las reglas y el espíritu de Scrum
- Ayudan a eliminar impedimentos que identifican el Equipo o el Dueño de Producto
- Hacen que la experiencia y el conocimiento estén disponibles

En Scrum, estas personas sustituyen el tiempo que antes empleaban como “niñeras” (asignando tareas, pidiendo informes de progreso y otras formas de micro-gestión) por tiempo como “gurú” o “sirviente” del Equipo (mentorización, coaching, ayudar a eliminar obstáculos o resolver problemas, proporcionar feedback constructivo y guiar el desarrollo de habilidades de los miembros de Equipo). En este cambio, los gerentes pueden tener que cambiar su estilo de gestión; por ejemplo, mediante el uso de preguntas socráticas para ayudar al equipo a descubrir la solución a un problema por sí mismos en lugar de simplemente decidiendo una solución y asignándosela al equipo.

Backlog de Producto

Cuando un grupo piensa en la transición a Scrum, antes de que comience el primer Sprint, es necesario un **Backlog de Producto**, una lista priorizada (ordenada 1, 2, 3...) de funcionalidades desde la perspectiva del cliente.

El Backlog de Producto existe (y evoluciona) durante toda la vida del producto; es la hoja de ruta del mismo (**Figura 2** y **Figura 3**). En cualquier momento, el Backlog de Producto es la visión única y definitiva de “todo lo que podría ser realizado en algún momento por el Equipo, en orden de prioridad”. Sólo existe un Backlog de Producto para cada producto, lo cuál significa que el Dueño de Producto debe tomar decisiones de prioridad sobre todo el abanico de opciones, en representación de los intereses de todos los stakeholders (incluido el Equipo).

Prioridad	Elemento	Detalles (URL Wiki)	Estimación Inicial	Nueva estimación en Sprint...					
				1	2	3	4	5	6
1	Como comprador, quiero poner un libro en el carrito de la compra (ver esquema de interfaz en el Wiki)	...	5						
2	Como comprador, quiero eliminar un libro del carrito de la compra	...	2						
3	Mejorar el rendimiento del procesamiento de transacciones (ver métricas objetivo en el Wiki)	...	13						
4	Investigar solución para mejorar velocidad de validación de tarjetas de crédito (ver objetivo de métricas de rendimiento den Wiki)	...	20						
5	Actualizar todos los servidores a Apache 2.2.3	...	13						
6	Diagnosticar y arreglar los errores del script de procesamiento de pedido (ID buqzilla 14823)	...	3						
7	Como comprador, quiero crear y guardar una lista de deseos	...	40						
8	Como comprador, quiero añadir y borrar artículos en mi lista de deseos	...	20						

Figura 2. El Backlog de Producto

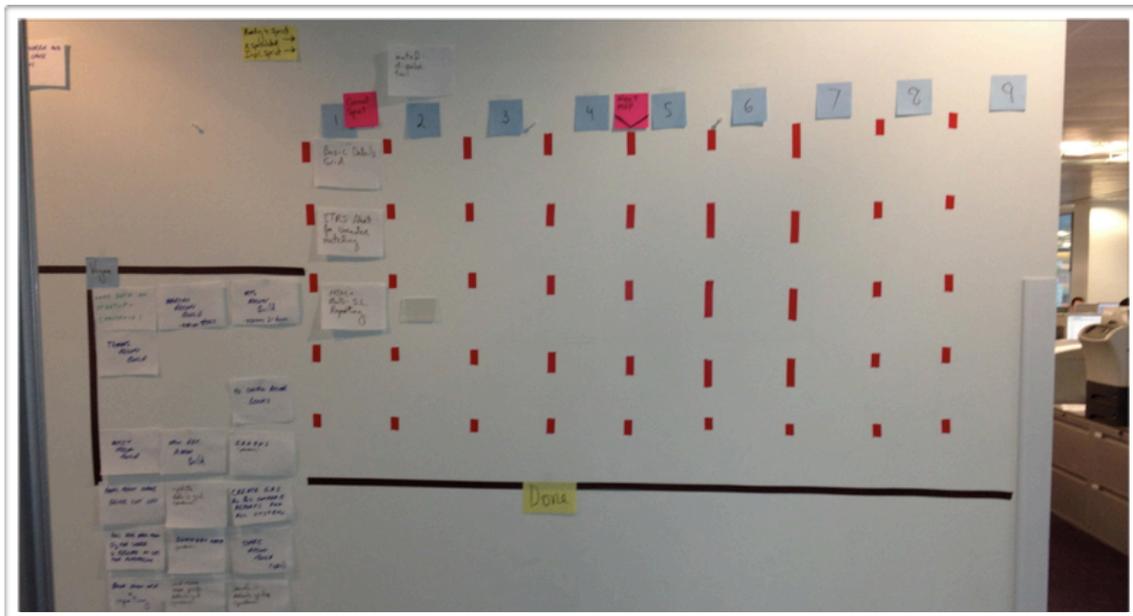


Figura 3. Visual Management: Elementos del Backlog de Producto en la pared

El Backlog de Producto incluye una variedad de elementos, principalmente nuevas funcionalidades para el cliente (“permitir a todos los usuarios colocar un libro en el carrito de la compra”), pero también objetivos *relevantes* de mejora de la ingeniería (por ejemplo, “re-escribir el sistema de C++ a Java”), objetivos de mejora (por ejemplo, “aumentar la velocidad de los tests”), trabajo de investigación (“investigar soluciones para validar tarjetas de crédito más rápidamente”) y,

posiblemente, defectos conocidos (“diagnosticar y corregir los errores del script de procesamiento de pedidos”) si sólo existen unos pocos problemas (un sistema con muchos defectos habitualmente tendrá un sistema separado de trazabilidad de defectos).

Los elementos del Backlog de Producto se articulan de cualquier manera que sea clara y sostenible. Al contrario de lo que se suele interpretar, el Backlog de Producto *no* contiene “historias de usuario”; contiene simplemente *elementos*. Estos elementos se pueden expresar en la forma de historias de usuario, casos de uso o cualquier otra aproximación a la toma de requisitos que el grupo considere útil. Pero sea cual sea la aproximación, la mayor parte de los elementos deberían enfocarse en proporcionar valor a los clientes.

Un buen Backlog de Producto es DEEP (profundo):

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

Estimado. Los elementos de que componen la siguiente versión del producto (release) deben estar estimados y, lo que es más, deberían revisarse y re-estimarse cada Sprint conforme vamos obteniendo nueva información. El Equipo proporciona al Dueño de Producto estimaciones de *esfuerzo* para cada elemento del Backlog del Producto, y quizás también proporcionan estimaciones de *riesgo técnico*. El Dueño de Producto y otros stakeholders proporcionan información sobre el valor de cada petición, lo cual puede incluir ingresos producidos, reducción de costes, riesgo de negocio, importancia para distintos stakeholders y otros datos.

Emergente. 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, obstáculos técnicos que aparecen, etcétera.

Priorizado. Los elementos del Backlog de Producto están priorizados de manera *ordenada*, 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, antes de que los riesgos te ataquen a ti*.

El modelo tradicional de desarrollo normalmente no subraya la necesidad de entregar el *mayor valor por el dinero*, pero en Scrum es un lema constante, y por tanto los Dueños de Producto deben aprender como evaluar el “valor de negocio”. Esto es algo en lo que los ScrumMasters pueden ayudar a los Dueños de Producto. ¿Qué significa “valor de negocio”? Algunas organizaciones emplean una estimación sencilla basada en “puntos de valor relativos” para cada elemento del Backlog de Producto que sintetiza una serie de “estimaciones intuitivas” sobre factores como ingresos producidos, reducción de coste, preferencias de los stakeholders, diferenciación de mercado etcétera. En algunas otras, los elementos son financiados cuando uno o más clientes pagan por su desarrollo, así que se usa la ganancia (a corto plazo) de dicho elemento como aproximación de su valor. Para otras organizaciones, este enfoque basado en valor por cada elemento es demasiado disperso o granular, por lo que aplican una estrategia basada en resultados de negocio más amplios (“incrementar las suscripciones un 10% para el 1 de Septiembre”); en esta estrategia el valor sólo se logra cuando múltiples elementos que contribuyen al resultado se entregan juntos. En estos casos, el Dueño de Producto debe definir el siguiente incremento del Producto Mínimo Viable (*Minimum Viable Product* o MVP).

En lo que respecta a las estimaciones de esfuerzo, una técnica comúnmente empleada es estimar los elementos en términos de su tamaño relativo (evaluando factores como esfuerzo, complejidad e incertidumbre) utilizando como unidad los “puntos de historia” o simplemente “puntos”.

Todo esto no son más que sugerencias: Scrum no define la técnica a emplear para expresar o priorizar los elementos del Backlog de Producto, como tampoco define la forma de estimarlos o las unidades a emplear.

Otra técnica comúnmente empleada en Scrum es llevar un cálculo de cuánto trabajo se completa al final de cada Sprint; por ejemplo, una media de 26 puntos finalizados por cada Sprint. Con esta información se puede planificar una fecha de lanzamiento de todos los elementos, o bien estimar cuántos elementos estarán listos en una fecha dada siempre que la media se mantenga y no hayan cambios relevantes. A esta media se la denomina “velocidad”. La velocidad se expresa en las mismas unidades que se empleen para estimar el tamaño de los elementos del Backlog de Producto.

Los elementos del Backlog de Producto pueden ser de muy distinto tamaño o esfuerzo. Los elementos más grandes se dividen en otros más pequeños durante las reuniones de Refinamiento del Backlog de Producto o las de Planificación de Sprint, y los elementos más pequeños pueden ser agrupados. Los elementos del Backlog de Producto que se van a desarrollar en los próximos Sprints deberían ser lo suficientemente pequeños y granulares para que el Equipo los entienda adecuadamente, permitiendo así que las estimaciones realizadas en las reuniones de Planificación de Sprint tengan sentido; esto es lo que se denomina un tamaño “manejable” (*actionable*).

Las mejoras de ingeniería relevantes que consuman mucho tiempo y dinero deberían estar en el Backlog de Producto, ya que podrían ser una opción de inversión de negocio a considerar, en última instancia, por un Dueño de Producto orientado a negocio. Hay que recalcar que en Scrum el Equipo tiene autoridad independiente en lo que concierne a cuántos elementos del Backlog de Producto entran en un Sprint, por lo que son libres de realizar pequeñas mejoras técnicas que se considerarán una parte del coste normal de desarrollo que es necesaria para que los desarrolladores puedan realizar su trabajo correctamente. No obstante, en cada Sprint la *mayoría* del tiempo del Equipo debería emplearse normalmente en objetivos del Dueño de Producto, no en tareas internas de ingeniería.

Uno de los mitos en torno a Scrum es que no permite escribir especificaciones detalladas; en realidad, depende del Dueño de Producto y del Equipo decidir cuánto detalle es necesario, y esto puede variar de un elemento del Backlog de Producto a otro dependiendo de las percepciones del Equipo y de otros factores. El objetivo es expresar en el menor espacio posible aquello que es importante o, en otras palabras, no describir todos los detalles posibles de un elemento sino clarificar lo que sea necesario entender y completarlo mediante un diálogo continuo entre el Equipo, el Dueño de Producto y los stakeholders. Los elementos de baja prioridad en el Backlog de Producto, en los que no vamos a trabajar hasta dentro de un tiempo, son normalmente de “grano grueso” (grandes, con mucho menor nivel de detalle). Los elementos de alta prioridad y grano fino, en los que típicamente vamos a trabajar pronto, tienden a tener un mayor nivel de detalle.

Definición de “Terminado”

El resultado de cada Sprint se denomina oficialmente un Incremento de Producto Potencialmente Entregable. Antes de comenzar el primer Sprint, el Dueño de Producto, el Equipo y el ScrumMaster deben revisar todo lo que hace falta para que un elemento del Backlog de Producto sea potencialmente entregable. Todas las actividades necesarias para entregar dicho producto deben estar incluidas en la definición de Potencialmente Entregable y por tanto deberían realizarse durante el Sprint.

Desafortunadamente, cuando los equipos comienzan a emplear Scrum ocurre muy a menudo que no son capaces de lograr un Incremento Potencialmente Entregable al final de cada Sprint. Muchas veces esto se debe a que el equipo no cuenta con suficiente automatización o no es suficientemente cross-funcional (por ejemplo, si los redactores técnicos no están aun incluidos en el equipo). Con el tiempo, el equipo debe mejorar de forma que sea capaz de realizar un Incremento de Producto Potencialmente Entregable al final de cada Sprint, pero para empezar es necesario que comprendan de qué son capaces actualmente. Esto queda registrado en la Definición de “Terminado”.

Antes del primer Sprint, el Dueño de Producto y el Equipo deben acordar una Definición de “Terminado”, que será un subconjunto de las actividades necesarias para crear un Incremento de Producto Potencialmente Entregable (en un buen equipo, será el mismo conjunto de actividades). El Equipo planificará su Sprint atendiendo a esta Definición de “Terminado”.

Un buen Dueño de Producto siempre querrá que la Definición de “Terminado” sea lo más cercana posible a Potencialmente Entregable, ya que esto incrementará la transparencia en el desarrollo y reducirá el *retraso* y el *riesgo*. El trabajo retrasado se denomina en ocasiones *trabajo sin terminar*.

Un Equipo Scrum mejora de forma continua, lo que se refleja en la extensión de su Definición de “Terminado”.

Planificación de Sprint

Resumen: Una reunión para preparar el Sprint, típicamente dividida en dos partes (la primera parte es el “qué” y la segunda es el “cómo”).

Participantes: Primera Parte: Dueño de Producto, Equipo, ScrumMaster. Segunda Parte: Equipo, ScrumMaster, Dueño de Producto (opcional, pero disponible para aclaraciones).

Duración: Cada parte se acota a una hora por cada semana de duración del Sprint.

Al principio de cada Sprint tiene lugar la reunión de **Planificación de Sprint**. Se divide en dos sub-reuniones diferentes, la primera de las cuales se denomina **Primera Parte de la Planificación de Sprint**.

En la **Primera Parte de la Planificación de Sprint**, 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. Habitualmente, estos elementos han sido bien analizados durante los Sprints anteriores (durante los Refinamientos de la Pila de Producto), por lo que durante esta reunión deberían realizarse únicamente aclaraciones menores de última hora. En esta reunión, el Dueño de Producto y el Equipo discuten los objetivos y el contexto para estos elementos de alta prioridad, proporcionando así al Equipo la visión del Dueño de Producto. La Primera Parte se centra en *qué* es lo que el Dueño de Producto quiere y *por qué* es necesario. Al final de la Primera Parte, el (siempre ocupado) Dueño de Producto puede abandonar la reunión, aunque *debe* permanecer disponible (por ejemplo, vía telefónica) durante la Segunda Parte.

En la Primera Parte, el Equipo y el Dueño de Producto pueden también formular un **Objetivo del Sprint**. Se trata de una declaración que resume la finalidad del Sprint y que, idealmente, tendrá un tema central. El Objetivo del Sprint también proporciona al Equipo flexibilidad sobre el alcance que finalmente entregarán, ya que aunque puedan tener que eliminar algunos elementos (dado que el Sprint está acotado en el tiempo), tendrían de todas formas que comprometerse a entregar algo tangible y “terminado” que esté en consonancia con el espíritu del Objetivo del Sprint.

¿Cómo de grandes deberían ser los elementos que se seleccionan para un Sprint? Cada elemento debería dividirse hasta ser lo suficientemente pequeño como para que pueda realizarse en un mucho menos de un Sprint completo. Una regla común es que un elemento debería ser suficientemente pequeño para ser completado por todo el Equipo en la cuarta parte de un Sprint o menos.

La Segunda Parte de la Planificación de Sprint se centra en *cómo* implementar los elementos que el Equipo decide incluir en el Sprint. El Equipo estima la cantidad de elementos que pueden completar al final del Sprint, comenzando desde la parte superior del Backlog de Producto (o, en otras palabras, empezando con los elementos que son de máxima prioridad para el Dueño de Producto) y continuando a partir de ahí en orden descendente. Esta es una práctica clave de Scrum: *el Equipo decide cuánto trabajo pueden completar, en lugar de que este les sea asignado por el Dueño de Producto*. Esto proporciona estimaciones más fiables, ya que el Equipo las realiza basándose en su propio análisis y planificación. Aunque el Dueño de Producto no tiene control sobre cuántos elementos incluye el equipo en el Sprint, sabe que los elementos que el Equipo selecciona son los de la parte superior del Backlog de Producto - es decir, aquellos elementos que ha marcado como más importantes. El Equipo tiene la posibilidad de proponer elementos de partes más bajas de la lista; esto suele ocurrir cuando el Equipo y el Dueño de Producto se dan cuenta de que algo de menor prioridad puede encajar fácil y apropiadamente con los elementos de máxima prioridad.

La reunión de Planificación de Sprint suele durar varias horas, pero nunca más de cuatro horas para un Sprint de dos semanas – el equipo está realizando una estimación seria para completar el trabajo, y debe meditar cuidadosamente para poder tener éxito. La Primera y Segunda Parte tienen una duración equivalente; para un Sprint de dos semanas, dos horas cada una como máximo.

Scrum no define exactamente cómo realizar la Segunda Parte de la Planificación de Sprint. Algunos equipos emplean la velocidad de los últimos Sprints como guía de cuánto deberían poder completar. Otros equipos emplean aproximaciones más detalladas para calcular en primer lugar su capacidad.

Cuando se emplea el enfoque basado en capacidad, el Equipo comienza la Segunda Parte de la Planificación de Sprint calculando cuánto tiempo tiene disponible cada miembro del Equipo para trabajos relacionados con el Sprint. La mayoría de los equipos estiman unas 4 a 6 horas disponibles al día para el Sprint – el resto del tiempo se emplea en gestionar el correo electrónico, pausas para almorzar, Facebook, reuniones y cafés. Una vez que se determina la capacidad, el Equipo debe estimar cuántos elementos del Backlog de Producto pueden desarrollar en este tiempo y como harán para desarrollarlos. Para ello suele comenzarse con una discusión en la pizarra. Una vez que todo el mundo comprende el diseño general, el Equipo descompone los elementos del Backlog de Producto en tareas concretas. Antes de escoger los elementos del Backlog de Producto, el Equipo puede concentrarse en generar tareas para un objetivo de mejora surgido de la última Retrospectiva. A continuación, el Equipo selecciona el primer elemento del Backlog de Producto – el de mayor prioridad para el Dueño de Producto – y continúan procesando elementos en orden hasta que estén “llenos”. Para cada elemento se crea una lista de trabajo que, o bien consiste en una descomposición en tareas del elemento del Backlog de Producto o, en el caso de que el elemento sea tan pequeño que apenas necesite un par de horas para realizarse, simplemente refleja el propio elemento. Esta lista de trabajo a realizar durante el Sprint se denomina el **Backlog del Sprint** (Figura 4 y Figura 5).

		Nueva estimación de esfuerzo pendiente del día...							
Elemento del Backlog	Tareas del Sprint	Voluntario	Estimación inicial de esfuerzo	1	2	3	4	5	6
Como comprador, quiero poner un libro en el carrito de la compra	Modificar base de datos		5						
	crear página Web (UI)		8						
	Crear página Web (lógica Javascript)		13						
	Escribir tests de aceptación automatizados		13						
	Actualizar web de ayuda al comprador		3						
	...								
Mejorar rendimiento de procesamiento de transacciones	Fusionar código DCP y completar tests a nivel de capa		5						
	Completar pedido de máquina para pRank		8						
	Cambiar DCP y lector para usar API http pRank		13						

Figura 4. Ejemplo de una forma de crear un Backlog de Sprint

Al final de la reunión de Planificación de Sprint, el Equipo establece un objetivo realista de lo que consideran que pueden entregar al final del Sprint. Tradicionalmente, a esto se le denominaba el Compromiso del Sprint – el Equipo se compromete a hacer todo lo que pueda para alcanzar dicho objetivo. Desafortunadamente, esto se ha malinterpretado frecuentemente más como una “promesa de sangre” que como un compromiso serio del Equipo de intentarlo por todos los medios. Para evitar esta confusión, al objetivo del Sprint se considera ahora una “estimación” que se proporciona al Dueño de Producto.

Scrum recomienda contar con trabajadores que tengan múltiples habilidades en lugar de los “puestos de trabajo” como el “tester” que solo hace pruebas. Dicho de otra forma, los miembros del Equipo deben “ir a por el trabajo” y ayudar en todo lo posible. Si hay demasiadas tareas de pruebas, entonces *todos* los miembros del equipo pueden ser de ayuda. Esto no implica que todo el mundo sea un generalista; sin duda, algunas personas tendrán más conocimientos de pruebas (como en todo), pero los miembros del Equipo trabajan juntos y aprenden nuevas habilidades unos de otros. Por ello, durante la creación y estimación de tareas en la Planificación de Sprint, no es necesario – ni apropiado – que la gente escoja todas las tareas que “pueden hacer mejor”. En su lugar, es mejor escoger las tareas de una en una - cuando sea el momento de escoger una nueva tarea – y considerar el escoger conscientemente tareas que implican un aprendizaje (quizás mediante el trabajo en pareja con un especialista). Esta es una de las razones para no pre-asignar tareas durante la Planificación de Sprint y, por el contrario, realizar esta asignación “conforme haga falta” durante el Sprint.

Dicho esto, hay *raras* ocasiones en las que *Juan* puede hacer una tarea concreta porque al resto les llevaría demasiado tiempo o les sería casi imposible aprender a hacerla – quizás Juan es la única persona con cierta habilidad artística para realizar un gráfico mientras que el resto de miembros del equipo no podrían dibujar un “monigote” ni aunque su vida dependiera de ello. En estas raras ocasiones – y si no son raras ni se van ir haciendo cada vez más raras conforme el Equipo aprende, entonces es que algo no funciona – puede ser necesario preguntar si es factible que todas las tareas de dibujo planificadas que *tienen* que ser realizadas por Juan quepan en el Sprint.

Muchos Equipos tienen su Backlog de Sprint en un tablón en la pared (habitualmente conocido como el **Tablón Scrum**) donde las tareas (escritas en notas Post-It) van moviéndose durante el Sprint a través de columnas denominadas “Pendiente”, “Trabajo en Curso” y “Terminado”. Ver Figura 5.



Figura 5. Visual Management – tareas del Backlog del Sprint en la pared

Uno de los pilares de Scrum es que, una vez que el Equipo ha generado un Objetivo del Sprint, cualquier añadido o cambio debe diferirse hasta el próximo Sprint. Esto significa que si a mitad de Sprint el Dueño de Producto decide que hay un nuevo elemento que quisiera que el Equipo desarrollase, no podrá realizar este cambio hasta el comienzo del próximo Sprint. Si surgen circunstancias externas que cambien las prioridades de forma significativa y esto significa que el equipo estaría perdiendo el tiempo si continua trabajando, el Dueño de Producto y el Equipo pueden abortar el Sprint. El Equipo se detiene y una nueva Planificación de Sprint da comienzo a un nuevo Sprint. La interrupción que esto introduce es importante; esto sirve como desincentivo para que el Equipo y el Dueño de Producto eviten recurrir a esta drástica solución.

Existe una gran influencia positiva que surge de proteger al equipo del cambio de objetivos durante el Sprint. En primer lugar, el Equipo puede trabajar con total certidumbre de que el objetivo no se verá alterado, lo cuál centra el esfuerzo del Equipo en lograr completarlo. En segundo lugar, hace que el Dueño de Producto deba ser disciplinado a la hora de pensar qué elementos prioriza en el Backlog de Producto y ofrece al Equipo para el Sprint.

Siguiendo estas reglas de Scrum, el Dueño de Producto logra dos cosas. En primer lugar, tiene la seguridad de saber que el Equipo se ha comprometido a hacer todo lo que pueda para completar un conjunto de elementos realista y concreto que ellos mismos han escogido. Con el tiempo, el Equipo puede ser muy bueno a la hora de estimar los elementos que puede escoger de forma realista. En Segundo lugar, el Dueño de Producto puede hacer todos los cambios que desee antes del comienzo del próximo Sprint. En ese punto es posible y aceptable añadir, eliminar, modificar o re-priorizar cualquier cosa. Aunque el Dueño de Producto no puede realizar cambios en los elementos seleccionados para desarrollo durante el Sprint actual, solo se encuentra a un Sprint o menos de poder hacer todos los cambios que quiera. Se acabó el estigma de los cambios – cambios de dirección, cambios de requisitos o simplemente cambio de idea – y puede que sea por esta razón que los Dueños de Producto son tan entusiastas de Scrum como cualquiera.

Scrum Diario

Resumen: actualización y coordinación entre los miembros del Equipo .

Participantes: Es necesario que asista el Equipo. El Dueño de Producto es opcional. El ScrumMaster normalmente está presente pero se asegura de que el equipo celebre uno.

Duración: Máximo 15 minutos

Una vez que el Sprint ha comenzado, el equipo acomete otra de las prácticas clave de Scrum: el **Scrum Diario**. Se trata de una corta reunión (15 minutos o menos) que tiene lugar todos los días a una hora convenida. Todo el Equipo asiste. Para que sea breve, se recomienda que todo el mundo permanezca de pie. Es la oportunidad para que el equipo sincronice su trabajo y se informen unos a

otros sobre los obstáculos. En el Scrum Diario los miembros del equipo, uno por uno, informan *al resto del equipo* de tres aspectos: (1) ¿Qué han realizado desde la última reunión? (2) ¿Qué estará realizado en la próxima reunión? (3) ¿Qué obstáculos han surgido? Nótese que el Scrum Diario no es una reunión de reporte a un gerente; es un momento para que un Equipo auto-organizado comparta lo que está ocurriendo para así poder coordinarse. Alguien anota los bloqueos, y el ScrumMaster es responsable de ayudar a los miembros del Equipo a resolverlos. Hay muy poca o ninguna discusión en profundidad durante el Scrum diario, el objetivo es *compartir* las respuestas a las tres preguntas; si hace falta discutir algo se hará justo a continuación de la reunión de Scrum Diario en una más reuniones paralelas de seguimiento, aunque en Scrum nadie está obligado a atender a dichas reuniones. Una reunión de seguimiento es un evento común donde alguno o todos los miembros del Equipo se adaptan a la información que han recibido en el Scrum Diario: en otras palabras, se trata de otro ciclo de inspección y adaptación. Para equipos que son nuevos en Scrum, se recomienda que *no* asistan gerentes u otras personas en posiciones que se perciban como de autoridad. Esto introduciría el riesgo de que el Equipo se sienta “monitorizado” – presionado para reportar progresos importantes todos los días (lo cuál es una expectativa poco realista), e inhibido para informar sobre problemas – y tendería a socavar la auto-gestión del Equipo e invitar a la microgestión. Sería más productivo para los stakeholders aproximarse al Equipo después de las reuniones y ofrecerse a ayudar con cualquier bloqueo que esté frenando el progreso del Equipo.

Registrando el progreso durante el Sprint

El Equipo Scrum es auto-gestionado, y para poder lograrlo debe saber cómo se está comportando. Todos los días, los miembros del Equipo actualizan su estimación del esfuerzo que queda para completar el trabajo actual del **Backlog del Sprint** (Figura 6). Es también común que alguien sume todo el esfuerzo pendiente del Equipo y lo registre en el **Diagrama de BurnDown del Sprint** (Figura 7 y Figura 8). 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 *burn-down* (ir “quemando” trabajo hacia abajo). Y aunque a veces tiene buen aspecto, muy a menudo no lo tiene; esta es la realidad del desarrollo de productos. Lo importante es que le muestra al equipo su progreso hacia el objetivo, no en términos de cuánto tiempo se ha *invertido* en el pasado (un hecho irrelevante de cara al *progreso*), sino en términos de cuanto trabajo *queda en el futuro* – cuanto separa al Equipo de su objetivo. 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.

Aunque el Diagrama de BurnDown puede ser creado y mostrado mediante el uso de una hoja de cálculo, muchos equipos consideran más efectivo mostrarlo en un papel colgado en una pared de su zona de trabajo y actualizarlo a mano; esta solución de “baja tecnología – alta interacción” es rápida, sencilla y generalmente más visible que un gráfico en un ordenador.

		Nuevas estimaciones de Esfuerzo al final del día...							
Elemento del Backlog	Tareas del Sprint	Voluntario	Estimación inicial de esfuerzo	1	2	3	4	5	6
Como comprador, quiero poner un libro en el carrito de la compra	Modificar base de datos	Sanjay	5	4	3	0	0	0	
	crear página Web (UI)	Jing	3	3	3	2	0	0	
	Crear página Web (lógica Javascript)	Tracy & Sam	2	2	2	2	1	0	
	Escribir tests de aceptación automatizados	Sarah	5	5	5	5	5	0	
	Actualizar web de ayuda al comprador	Sanjay & Jing	3	3	3	3	3	0	
	...								
Mejorar rendimiento de procesamiento de transacciones	Fusionar código DCP y completar tests a nivel de capa		5	5	5	5	5	5	
	Completar pedido de máquina para pRank		3	3	8	8	8	8	
	Cambiar DCP y lector para usar API http pRank		5	5	5	5	5	5	
...									
Total			50	4	4	4	4	3	
				9	8	4	3	4	

Figura 6. Actualizaciones de trabajo pendiente en el Backlog del Sprint

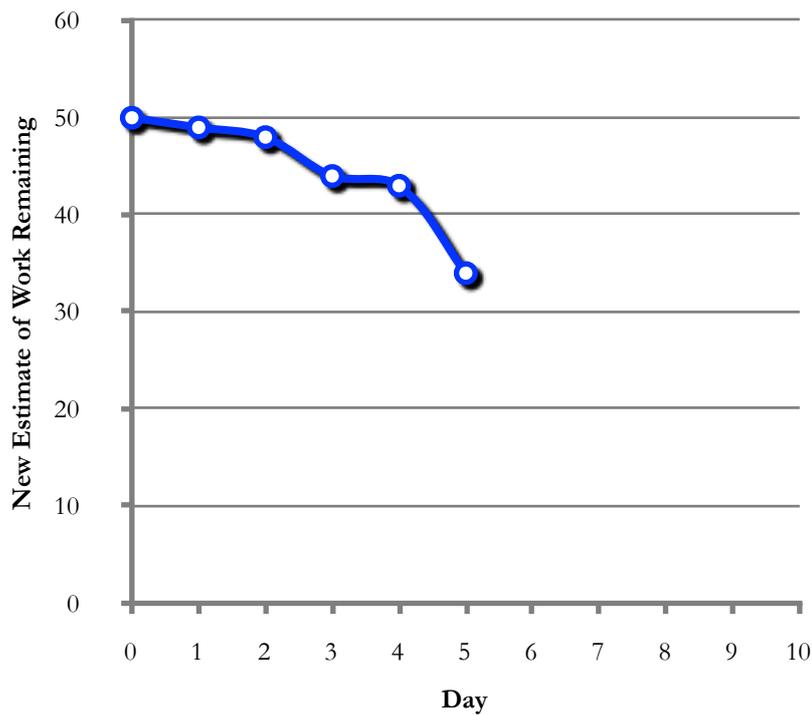


Figura 7. Diagrama de Burndown del Sprint

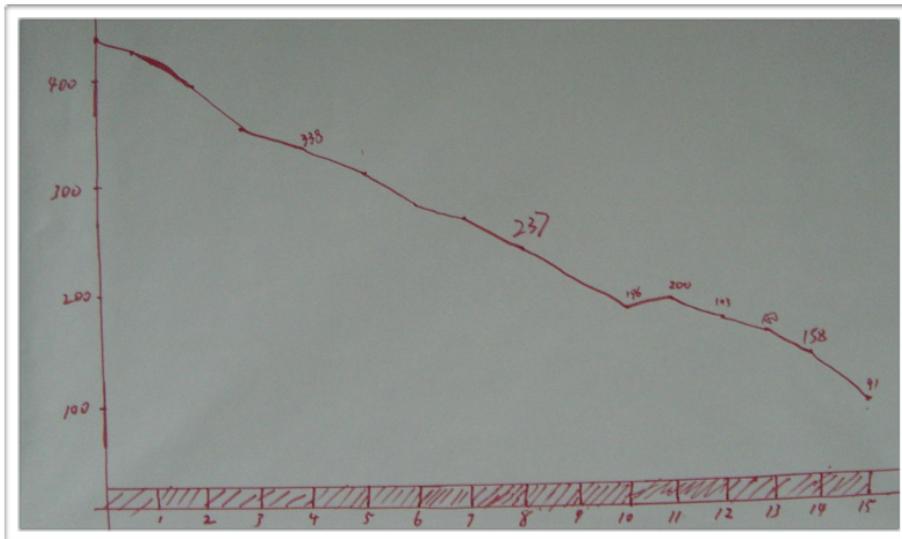


Figura 8. Visual Management: Diagrama de BurnDown del Sprint dibujado a mano

Refinamiento del Backlog de Producto

Resumen: Dividir elementos grandes, analizar elementos, re-estimar y re-priorizar para los próximos Sprints

Participantes: Equipo; el Dueño de Producto asistirá si es un experto que puede clarificar los detalles necesarios para el refinamiento, en caso contrario puede asistir sólo a una parte de la reunión para establecer un objetivo o re-priorizar; otras personas que entiendan los requisitos y puedan ayudar al Equipo; el ScrumMaster asistirá a las primeras reuniones para ayudar al Equipo a ser efectivo, más adelante puede no asistir.

Duración: Normalmente no más del 10% de la capacidad del Equipo para el Sprint, aunque podría ser más si hay elementos “duros de analizar”. Por ejemplo, en un Sprint de dos semanas quizás se emplee un día en refinamiento.

Una de las recomendaciones menos conocidas pero más valiosas de Scrum es que cierto porcentaje del Sprint el Equipo al completo debería dedicarse a refinar (o “acicalar” – “*grooming*”) el Backlog de Producto para preparar los próximos Sprint. Esto incluye analizar los requisitos al detalle, dividir elementos grandes en otros más pequeños, estimar nuevos elementos y re-estimar elementos existentes. Scrum no dice nada al respecto de cómo realizar este trabajo, pero una técnica habitual es realizar un taller específico cerca de la mitad o el final del Sprint, de forma que el Equipo, el Dueño de Producto y otros stakeholders puedan dedicarse a este trabajo sin interrupciones.

Esta labor de refinamiento *no se realiza* sobre los elementos del Sprint actual; es para los elementos futuros, típicamente para los que se realizarán en los próximos uno o dos Sprints. Mediante esta práctica, el proceso de Planificación de Sprint se hace relativamente sencillo ya que el Dueño de Producto y el Equipo Scrum comienzan la planificación con un conjunto de elementos claro, bien analizado y cuidadosamente estimado. Un síntoma de que no se están realizando estos talleres de refinamiento (o que no se están realizando bien) es que en la Planificación de Sprint se realizan muchas preguntas significativas, se descubren cosas, existe confusión o parece incompleta; el trabajo de planificación entonces se desborda al propio Sprint, lo cuál no es deseable.

Revisión del Sprint

Resumen: Inspección y adaptación respecto al incremento de funcionalidad del producto

Participantes: Equipo, Dueño de Producto, ScrumMaster. Otros stakeholders que sean adecuados, invitados por el Dueño de Producto.

Duración: acotado a una hora por cada semana del Sprint

Una vez que el Sprint termina, se realiza una **Revisión del Sprint**. En esta reunión están el Dueño de Producto, el Equipo y el ScrumMaster, además de clientes, usuarios, stakeholders, expertos, ejecutivos y cualquier otro interesado. Para un Sprint de dos semanas tendrá una duración máxima de dos horas. Cualquier persona presente puede realizar preguntas y proporcionar opiniones.

Con frecuencia, a la revisión se la denomina erróneamente “demo”, pero esto no refleja la auténtica intención de esta reunión. Una idea clave de Scrum es *inspección y adaptación*. Observar y aprender de lo que está ocurriendo y evolucionar basándonos en el feedback en ciclos repetitivos. La revisión del Sprint es una actividad de inspección y adaptación del *producto*. Es un momento para que el Dueño de Producto comprenda lo que está ocurriendo con el producto y el Equipo (es decir, una revisión del Sprint); y para que el Equipo entienda qué está pasando con el Dueño de Producto y el mercado. Consecuentemente, un elemento clave de la Revisión es una *conversación* profunda entre el Equipo y el Dueño de Producto para entender la situación, pedir consejo, etcétera. La Revisión indudablemente incluye probar el software que el Equipo ha estado desarrollando durante el Sprint, pero si el foco de la Revisión es únicamente mostrar el producto en lugar de mantener una conversación, existe un desequilibrio.

La parte de “software funcionando” de la Revisión de Sprint no es una “presentación” que el Equipo realiza – no deberían haber diapositivas ni nada similar. Debería ser una inspección real del auténtico software corriendo “en vivo”, por ejemplo en entornos de desarrollo de pruebas. Deberían haber uno o más ordenadores en la sala de la Revisión en los que todo el mundo pudiera inspeccionar y usar el software. Es preferible una sesión activa en la que usuarios reales y el Dueño de Producto realicen interacciones reales con el software que una demostración pasiva por parte del Equipo.

No deberían emplearse más de 30 minutos en preparar una Revisión de Sprint, si no es así es probable que exista algún problema.

Retrospectiva del Sprint

Resumen: inspección y adaptación relativas al proceso y el entorno.

Participantes: Equipo, ScrumMaster, Dueño de Producto (opcional) . Otros stakeholders pueden ser invitados por el Equipo, en caso contrario no se les permite asistir.

Duración: acotada a 45 minutos por semana de Sprint.

La Revisión del Sprint consiste en una inspección y adaptación del *producto*. La **Retrospectiva del Sprint**, que se realiza justo a continuación de la Revisión, consiste en una inspección y adaptación del *proceso y el entorno*. Se trata de una oportunidad para que el Equipo discuta qué está funcionando y

qué no está funcionando, y acordar los cambios a probar. A veces el ScrumMaster puede actuar como facilitador efectivo de la Retrospectiva, pero podría ser mejor encontrar a alguien neutral de fuera del entorno del Equipo para facilitar esta reunión. Una buena estrategia para los ScrumMasters es que unos faciliten las retrospectivas de otros, lo que permite polinización cruzada entre Equipos.

Existen muchas técnicas para realizar una Retrospectiva de Sprint, y el libro *Agile Retrospectives* (Derby, Larsen 2006) proporciona un catálogo muy útil de algunas de ellas.

Muchos equipos realizan Retrospectivas que solo se centran en *problemas*, y esto es muy negativo. Puede hacer que las personas consideren la Retrospectiva como un evento negativo o deprimente. En vez de eso, hay que asegurarse de que parte de toda Retrospectiva se enfoca en fortalezas o en aspectos positivos; hay varios libros sobre “*indagación apreciativa*” (“*appreciative inquiry*”) que ofrecen consejos más detallados.

Las Retrospectivas que usan siempre la misma técnica de análisis pueden volverse aburridas; por tanto, es recomendable introducir diferentes técnicas con el tiempo.

Comenzando el siguiente Sprint

A continuación de la Revisión del Sprint, el Dueño de Producto puede actualizar el Backlog de Producto con cualquier nueva idea – añadir nuevos elementos, eliminar los que son obsoletos o revisar los actuales. El Dueño de Producto es responsable de asegurarse de que estos cambios se reflejan en el Backlog de Producto. En la Figura 9 se muestra un ejemplo de un Backlog de Producto actualizado.

Prioridad	Elemento	Detalles (URL Wiki)	Estimación Inicial	Nueva estimación en Sprint...				
				1	2	3	4	5
1	Como comprador, quiero poner un libro en el carrito de la compra (ver esquema de interfaz en el Wiki)	...	5	0	0	0		
2	Como comprador, quiero eliminar un libro del carrito de la compra	...	2	0	0	0		
3	Mejorar el rendimiento del procesamiento de transacciones (ver métricas objetivo en el Wiki)	...	13	13	0	0		
4	Investigar solución para mejorar velocidad de validación de tarjetas de crédito (ver objetivo de métricas de rendimiento den Wiki)	...	20	20	20	0		
5	Actualizar todos los servidores a Apache 2.2.3	...	13	13	13	13		
6	Diagnosticar y arreglar los errores del script de procesamiento de pedido (ID bugzilla 14823)	...	3	3	3	3		
7	Como comprador, quiero crear y guardar una lista de deseos	...	40	40	40	40		
8	Como comprador, quiero añadir y borrar artículos en mi lista de deseos	...	20	20	20	20		
...				
			537	580	570	500		

Figura 9. Backlog de Producto actualizado

No hay tiempo muerto entre los Sprints – normalmente los Equipos pasan de la Retrospectiva una tarde a la siguiente Planificación de Sprint la mañana siguiente (o tras el fin de semana).

Uno de los principios del desarrollo Ágil es el “paso sostenible”, y solo a base de trabajar regularmente un número de horas razonable pueden los Equipos mantener este ciclo de forma indefinida. La productividad crece con el tiempo gracias a la evolución de las prácticas del Equipo y a la eliminación de los impedimentos a su productividad, no mediante el exceso de trabajo o comprometiendo la calidad.

Los Sprints continúan hasta que el Dueño de Producto decide que el producto está listo para lanzar (*release*). La visión que tiene Scrum de la perfección es que el producto está listo para ser potencialmente entregado al final de cada Sprint, lo que implica que no hay pendiente trabajo de preparación o “empaquetado” como por ejemplo documentación o pruebas. Esto implica que *todo*

está completamente *terminado* al final de cada Revisión del Sprint. Aun así, muchas organizaciones cuentan con prácticas de desarrollo, herramientas e infraestructuras débiles y no pueden alcanzar esta visión de la perfección, por lo que será necesario un “Sprint de Lanzamiento” para realizar este trabajo pendiente. Cuando se necesita un “Sprint de Lanzamiento”, se considerará un mal necesario, y el deber de la organización será mejorar sus prácticas hasta que esto no sea necesario nunca más.

Gestionando los Lanzamientos

Una pregunta que surge a veces es cómo se pueden realizar planificaciones de lanzamiento a largo plazo en un modelo iterativo. Hay dos casos a considerar: (1) un nuevo producto en su primera versión o (2) una nueva versión de un producto existente.

En el caso de un nuevo producto, o de *un producto existente en el que acabamos de adoptar Scrum*, existe la necesidad de realizar un refinamiento del Backlog de Producto antes del primer Sprint en el que el Dueño de Producto y el Equipo den forma a un Backlog de Producto correcto. Esto puede durar unos días o una semana, e incluirá un taller (denominado a veces de Creación del Backlog de Producto o de Planificación de Lanzamiento – Planificación de *Release*), cierta cantidad de análisis detallado de los requisitos y una estimación de todos los elementos que se incluirán en el primer lanzamiento.

Sorprendentemente, en Scrum no debería existir la necesidad de una planificación especial o extensa si se trata de un producto existente con un Backlog de Producto establecido. ¿Por qué? Porque el Dueño de Producto y el Equipo deberían estar realizando Refinamientos del Backlog de Producto todos los Sprints (dedicando el 10% del tiempo del mismo), preparándose de forma continua para el futuro. Este modo de *desarrollo continuo de producto* obvia la necesidad de fases puntuales y drásticas de preparación-ejecución-cierre que se suelen ver en los ciclos de desarrollo secuencial tradicionales.

Durante el taller inicial de Refinamiento del Backlog de Producto, y durante los subsiguientes Refinamientos de cada Sprint, el Equipo y el Dueño de Producto harán la planificación de lanzamiento y el refinamiento de las estimaciones, prioridades y contenidos conforme vayan aprendiendo.

Algunos lanzamientos están marcados por una fecha; por ejemplo, “lanzaremos la versión 2.0 de nuestro proyecto en la feria del 10 de Noviembre”. En estas situaciones, el Equipo hará todos los Sprints (y desarrollará todas las funcionalidades) posibles en el tiempo disponible. Otros productos requieren el desarrollo de un mínimo de funcionalidades antes de que podamos considerarlos terminados y el producto no se lanzará hasta que se cumplan estos requisitos, independientemente de lo que se tarde. Dado que Scrum requiere producir código potencialmente entregable al final de cada Sprint, el Dueño de Producto puede decidir realizar lanzamientos intermedios que permitan al cliente obtener beneficios de las partes completas lo antes posible.

Dado que no es posible saberlo todo de antemano, el foco es crear y refinar un plan para establecer una dirección general del lanzamiento, y clarificar como se realizarán las decisiones compromiso (por ejemplo, de alcance frente a fechas). Se puede ver esto como si la hoja de ruta (*roadmap*) nos guiase hacia un destino; qué carreteras tomemos y qué decisiones tomemos durante el viaje pueden establecerse por el camino.

El destino es más importante que el viaje.

La mayoría de los Dueños de Producto escogen una estrategia de lanzamiento. Por ejemplo, pueden decidir una fecha de lanzamiento y trabajar con el equipo para estimar cuántos elementos del Backlog de Producto pueden completarse para dicha fecha. En situaciones en las que se requiere un compromiso de “precio cerrado / fecha cerrada / alcance cerrado” – por ejemplo, en desarrollos por contrato – uno o más de estos parámetros pueden contar con un “margen” (*buffer*) predefinido que permita absorber cierta cantidad de incertidumbre y cambios. A este respecto, Scrum no es distinto de otras estrategias.

Enfoque a Aplicación o Producto

Para aplicaciones o productos – tanto para el mercado como para uso interno – Scrum aparta a las organizaciones del antiguo modelo centrado en *proyectos* hacia un modelo enfocado en *desarrollo continuo de aplicación/producto*. Ya no existe un proyecto con un comienzo, un transcurso y un cierre. Y por tanto, no existe el jefe de proyecto tradicional. En su lugar, existe simplemente un Dueño de Producto estable y un Equipo auto-gestionado a largo plazo que colabora en una serie “infinita” de

Sprints de duración establecida hasta que el producto o la aplicación se retira. Toda la gestión de “proyecto” necesaria se desempeña por el Equipo y por el Dueño de Producto – que es un cliente interno de negocio o es alguien de Gestión de Producto. No es gestionado por un Director de Tecnología o alguien de una Oficina de Proyectos. Scrum también puede utilizarse para *proyectos* que constituyan una iniciativa puntual (en lugar de la creación o evolución de aplicaciones de largo ciclo de vida); incluso en estos casos, el Equipo y el Dueño de Producto realizan las labores de gestión del proyecto.

¿Qué ocurre si no hay suficiente trabajo en una o más aplicaciones como para justificar la existencia de un Equipo a largo plazo para cada una? En estos casos, un solo Equipo estable a largo plazo puede desarrollar elementos de una aplicación en un Sprint y de otra en el Sprint siguiente. En estas situaciones los Sprints son frecuentemente más cortos, por ejemplo de una semana.

Ocasionalmente, no hay suficiente trabajo ni siquiera para la solución anterior, y el Equipo podría tener que desarrollar elementos de *varias* aplicaciones durante el mismo Sprint; en cualquier caso, hay que ser cuidadoso con esta solución, ya que puede desembocar en una multitarea improductiva entre múltiples aplicaciones. Un lema recurrente de productividad en Scrum es que el Equipo se *centra* en un producto o aplicación en cada Sprint.

Retos habituales

Scrum no es sólo un conjunto concreto de prácticas –se trata más bien, y de forma más importante, de un marco de trabajo que proporciona transparencia, y un mecanismo de “inspección y adaptación”. Scrum funciona a base de hacer visibles las disfunciones e impedimentos que impactan la efectividad del Equipo y del Dueño de Producto de forma que puedan ser solucionados. Por ejemplo, el Dueño de Producto puede desconocer el mercado, las funcionalidades o cómo estimar su valor de negocio relativo. O el Equipo puede carecer de la habilidad necesaria en la estimación del trabajo o en el desarrollo.

El marco de trabajo Scrum revelará rápidamente estas debilidades. Scrum no soluciona los problemas de desarrollo; los hace dolorosamente visibles, y proporciona un marco de trabajo para que las personas exploren formas de solucionar problemas en ciclos cortos y mediante pequeños experimentos de mejora.

Supongamos que el Equipo no logra entregar al final de su primer Sprint todo lo que estimaron debido a que cuentan con poca habilidad para analizar y estimar tareas. Para el Equipo, esto puede parecer un fracaso. Pero en realidad, esta experiencia es un primer paso necesario para ser más realistas y reflexivos en el futuro respecto a las estimaciones. Este patrón – Scrum haciendo visible las disfunciones del Equipo y permitiéndoles hacer algo al respecto – es el mecanismo básico que produce los beneficios más importantes para el Equipo que emplea Scrum.

Un error común cuando nos enfrentamos a una práctica de Scrum que supone un reto, es intentar cambiar Scrum. Por ejemplo, los equipos que tienen problemas para entregar pueden decidir hacer la duración del Sprint flexible de forma que nunca se queden sin tiempo – y, gracias a ello, asegurarse de que nunca tienen que aprender como realizar un mejor trabajo de estimación y de gestión de su tiempo. En esta línea, y sin la ayuda y el soporte de un ScrumMaster experimentado, las organizaciones pueden transformar Scrum en una imagen de sus propias debilidades y disfunciones, socavando los auténticos beneficios que Scrum les ofrece: hacer visible lo bueno y lo malo y dar a la organización la posibilidad de elevarse a un nivel superior.

Otro error común es asumir que una práctica está prohibida o no recomendada simplemente porque Scrum no la requiere de forma específica. Por ejemplo, Scrum no exige que el Dueño de Producto establezca una estrategia a largo plazo para su producto, ni requiere que los ingenieros busquen el consejo de otros ingenieros más experimentados sobre problemas técnicos complejos. Scrum deja a las personas involucradas tomar las decisiones adecuadas; y en la mayoría de los casos, estas dos prácticas (y muchas otras) son muy recomendables.

Otra situación con la que tener cuidado se da cuando los gerentes imponen Scrum a sus Equipos. Scrum trata de dar al Equipo espacio y herramientas para que se gestione por si mismo, y que esto surja de un mandato desde arriba no es la mejor receta para el éxito. Una aproximación mejor con la que comenzar sería que el equipo oiga hablar sobre Scrum a otro compañero o un gerente, obtenga formación completa en un curso profesional y entonces tome como Equipo la decisión de seguir firmemente estas prácticas durante un tiempo determinado, al final del cual el equipo evaluará su experiencia y decidirá si desea continuar. Las buenas noticias son que, aunque el primer Sprint suele suponer muchos retos para el Equipo, los beneficios de Scrum tienden a ser visibles al final del mismo, lo cual lleva a muchos equipos nuevos en Scrum a exclamar “¡Scrum es duro, pero sin lugar a dudas es mejor que lo que estábamos haciendo antes!”

Apéndice A: Lecturas Adicionales

Hay mucho material publicado sobre Scrum. En esta sección de referencia nos gustaría señalar algunos recursos adicionales disponibles on-line y un par de libros.

Material on-line:

- The Lean Primer - An introduction to Lean Thinking, an important influence to Scrum. <http://www.leanprimer.com>
- The Distributed Scrum Primer - Additional tips for teams who aren't co-located. <http://www.goodagile.com/distributedscrumprimer/>
- The ScrumMaster Checklist - A list of question that good ScrumMasters use. <http://www.scrummasterchecklist.org/>
- Feature Team Primer - Scaling Scrum with Feature Teams, <http://www.featureteams.org>
- The Agile Atlas - Core Scrum. ScrumAlliance description of Scrum. <http://agileatlas.org/atlas/scrum>
- Scrum Guide - Scrum.org description of Scrum. <http://www.scrum.org/Scrum-Guides>
- Agile Contracts Primer - How to make Scrum-friendly contracts. <http://www.agilecontracts.org/>

Libros:

- Leading Teams - Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum - Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum - Craig Larman, Bas Vodde
- Agile Project Management with Scrum - Ken Schwaber
- Succeeding with Agile: Software Development using Scrum - Mike Cohn

Apéndice B: Terminología

Backlog de Producto

Una lista priorizada de requisitos con la estimación de tiempos necesarios para convertirlos en funcionalidades completas del producto. Las estimaciones son más precisas conforme el elemento está priorizado más arriba en el Backlog de Producto. La lista es emergente, y cambia conforme lo hacen las condiciones de negocio o la tecnología.

Backlog de Sprint

Una lista del trabajo del Equipo para un Sprint. Usualmente se encuentra dividida en un conjunto de tareas más detalladas. La lista emerge durante la Planificación de Sprint y puede ser actualizada durante el Sprint mediante la eliminación de elementos o la adición de nuevas tareas conforme sea necesario. Cada tarea del Backlog del Sprint será objeto de seguimiento durante el Sprint y mostrará el esfuerzo pendiente para terminarla.

Burn Down

La tendencia de trabajo pendiente durante el tiempo de un Sprint, un lanzamiento o un producto. El origen de los datos es el Backlog de Sprint y el Backlog de Producto, y se traza el trabajo pendiente en el eje vertical y el periodo de tiempo (días del Sprint, o Sprints) en el eje horizontal.

Dueño de Producto

La persona responsable de gestionar el Backlog de Producto para maximizar el valor del producto. El Dueño de Producto es responsable de representar las necesidades de todo el que tiene interés en el proyecto y en el producto resultante.

Equipo

Un grupo de personas cross-funcional que es responsable de gestionarse a sí mismo para desarrollar un incremento de producto cada Sprint.

Equipo de Desarrollo

Otro nombre para el rol del Equipo.

Elemento del Backlog de Producto

Requisitos funcionales, no funcionales y otras labores, priorizados en orden de importancia para el negocio y de dependencia, y estimados. La precisión de la estimación depende de la prioridad y granularidad del elemento del Backlog de Producto, de forma que los elementos de máxima prioridad que pueden ser seleccionados para el próximo Sprint tendrán la máxima granularidad y precisión.

Estimación de Trabajo Pendiente (Elementos del Backlog del Sprint)

El número de horas que un miembro del equipo estima que quedan hasta terminar cualquier tarea. Esta estimación se actualiza al final de cada día en el que se trabaje en la tarea del Backlog del Sprint. Se estima el total de esfuerzo pendiente, independientemente del número de personas que estén realizando dicho trabajo.

Incremento

Funcionalidad de producto desarrollada por el Equipo durante cada Sprint que es potencialmente entregable o utilizable por los stakeholders del Dueño de Producto.

Incremento de Funcionalidad de Producto Potencialmente Entregable

Una porción completa del producto global que podría ser utilizada por los stakeholders del Dueño de Producto si decidieran implementarla.

Reunión de Planificación de Sprint

Una reunión acotada a cuatro horas (para Sprints de dos semanas) que inicia cada Sprint. La reunión se divide en dos partes, también acotadas a dos horas cada una. Durante la primera parte, el Dueño de Producto presenta los elementos de mayor prioridad del Backlog de Producto al Equipo. El Equipo y el Dueño de Producto colaboran para ayudar al Equipo a determinar qué cantidad del Backlog de Producto pueden convertir en funcionalidad durante el próximo Sprint. Durante la segunda parte, el Equipo planifica cómo lograrlo mediante el diseño y descomposición del trabajo, de forma que comprendan como alcanzarán el objetivo del Sprint.

Reunión de Retrospectiva de Sprint

Una reunión facilitada por el ScrumMaster en la que el Equipo al completo evalúa el Sprint que acaba de concluir y determina qué podría cambiar de forma que el próximo Sprint fuera más agradable y productivo.

Reunión de Revisión de Sprint

Una reunión acotada a dos horas (para un Sprint de dos semanas) que tiene lugar al final de cada Sprint y en la que el Equipo colabora con el Dueño de Producto y otros stakeholders inspeccionando el resultado del Sprint. Normalmente comienza con una revisión de los elementos del Backlog de Producto que se han completado, una discusión de las oportunidades, limitaciones y riesgos existentes, y un debate sobre qué sería lo mejor que podrían hacer a continuación (lo cual potencialmente provocará cambios en el Backlog de Producto). Solo pueden mostrarse funcionalidades de producto que estén completadas.

Scrum

No es un acrónimo, sino que se refiere a la jugada de rugby para volver a poner en juego un balón (*melé*).

Scrum Diario

Una pequeña reunión que el Equipo mantiene todos los días en la que los miembros del Equipo inspeccionan y sincronizan su trabajo, su progreso e informan de los impedimentos al ScrumMaster para su eliminación. Tras el Scrum Diario pueden organizarse reuniones de seguimiento para adaptarse al trabajo pendiente u optimizar el Sprint.

ScrumMaster

La persona responsable del proceso Scrum, su correcta implementación y la obtención del máximo beneficio de la misma.

Sprint

Una iteración o ciclo repetitivo de trabajo semejante, que produce un incremento de producto o sistema. No mayor de un mes, y usualmente mayor de una semana. La duración es fija a lo largo de todo el trabajo, y todos los equipos que trabajan en el mismo sistema o producto usan la misma duración de ciclo.

Stakeholder

Alguien con interés en el resultado del proyecto, bien porque lo ha financiado, lo utilizará o será afectado por él.

Tarea del Backlog de Sprint

Una de las tareas que el Equipo o un miembro del Equipo considera necesarias para convertir los elementos escogidos del Backlog de Producto en funcionalidad del sistema.

Terminado

Completado según el criterio de todas las partes que se rigen por los estándares, convenciones y reglas de la organización. Cuando se dice que algo está “Terminado” en la reunión de Revisión del Sprint, debe ser de conformidad con la definición acordada.

Time box

Un periodo de tiempo que no puede excederse y dentro del cuál se desarrolla una reunión o evento. Por ejemplo, un Scrum Diario tiene un *timebox* de quince minutos y finaliza al final de esos quince minutos sea como sea. Las reuniones pueden durar menos del timebox. Los Sprints duran exactamente ese tiempo.