

INFORMACIÓN BÁSICA DE SCRUM (THE SCRUM PRIMER)

Por Pete Deemer

Gabrielle Benefield

Craig Larman

Bas Vodde

Versión 1.1

Traducción de Leo Antoli

Agile-Spain



Certified Scrum Training Worldwide | www.ScrumTI.com

Esta obra tiene licencia *Creative Commons Reconocimiento-No comercial-Sin obras derivadas 3.0*.
Para ver la licencia en español: <http://creativecommons.org/licenses/by-nc-nd/3.0/es/>
Para ver la licencia en inglés y otros idiomas: <http://creativecommons.org/licenses/by-nc-nd/3.0/us/>
O manda una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.



Nota al lector: Hay muchas descripciones concisas de Scrum en Internet, y este texto básico quiere proporcionar el siguiente nivel de detalle. Este texto no está pensado como lo único necesario para aprender Scrum; es recomendable que los equipos que estén pensando en usar Scrum tengan alguno de los libros de Ken Schwaber, *Agile Project Management with Scrum* o *Agile Software Development with Scrum*, y que examinen las opciones de cursos y *coaching* disponibles; se puede encontrar más información en www.scrumalliance.org. Agradecemos a Ken Schwaber, Jeff Sutherland y Mike Cohn su generosa contribución.

© 2009 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

Desarrollo tradicional de software

La forma tradicional de construcción de software que usan empresas de todos los tamaños es el ciclo de vida secuencial conocido normalmente como “en cascada”. Aunque hay muchas variantes (como el modelo en V), normalmente empiezan con una fase de planificación detallada donde se piensa cuidadosamente en el producto final, se diseña y documenta en gran detalle. Se determinan las tareas necesarias para ejecutar el diseño, y el trabajo es organizado usando herramientas como diagramas de Gantt y aplicaciones como Microsoft Project. El equipo hace la estimación del tiempo de desarrollo sumando las estimaciones detalladas de cada uno de los pasos individuales. El equipo empieza a trabajar una vez que todos los interesados en el proyecto han revisado el plan concienzudamente y han dado su aprobación. Los miembros del equipo van completando su parte especializada del trabajo y se lo pasan a los siguientes, parecido a una línea de producción. Cuando se termina el trabajo, se manda al grupo de *testing* (a veces llamado grupo de calidad o de QA, “*Quality Assurance*”), que completan las pruebas antes de que el producto llegue al cliente. Durante todo el proceso se siguen estrictos controles para detectar desviaciones sobre el plan para asegurarse de que el producto es realmente el que se diseñó.

Este enfoque tiene ventajas e inconvenientes. La principal ventaja es su lógica aplastante — piensa antes de construirlo, escribe todo, sigue un plan, y mantén todo lo más organizado posible. Solo tiene un pequeño inconveniente: participan personas.

Por ejemplo, este enfoque implica que todas las buenas ideas se nos tienen que ocurrir al principio del ciclo de entrega, que es cuando podemos incorporarlas al plan. Pero como todos sabemos, las buenas ideas pueden aparecer a lo largo de todo el proceso —en el comienzo, en el medio, y a veces incluso el día antes del lanzamiento, y un proceso que no permita el cambio coartará esta innovación. Con el ciclo de vida en cascada una gran idea tardía no es una bendición, es una amenaza.

El ciclo de vida en cascada pone gran énfasis en escribir todo como el principal método de comunicación de información crítica. La asunción razonable es que si puedo anotar en papel todo lo que tengo en la cabeza, será lo más fiable para pasárselo a los demás miembros del equipo; además, si está en papel, hay una prueba tangible de que he hecho mi trabajo. La realidad, sin embargo, es que la mayoría de las veces nadie lee los documentos detallados de requisitos de 50 páginas. Y *cundo se leen*, los malentendidos se van acumulando. Un documento escrito es un dibujo incompleto de mis ideas; cuando tú lo lees, te creas otra abstracción, que ahora está a dos pasos de distancia de lo que *yo creo* que significa lo que dije en ese momento. No es sorprendente que haya graves problemas de entendimiento.

Otra cosa que ocurre cuando hay personas por medio es el momento “ajá” — la primera vez que ven el producto real. Inmediatamente se les ocurren 20 maneras diferentes en las que se podría haber hecho mejor. Desafortunadamente, estas ideas valiosas ocurren al final del ciclo de entrega, cuando los cambios son difíciles y perjudiciales — en otras palabras, cuando hacer las cosas correctas es más costoso, al menos usando un método tradicional.

Las personas no somos buenas prediciendo el futuro. Por ejemplo, tu competidor anuncia algo inesperado. Problemas técnicos no previstos fuerzan un cambio de dirección. Es más, la gente es especialmente mala planificando cosas inciertas en el futuro lejano — adivinar hoy lo que harás durante la semana de dentro de ocho meses es ciencia ficción. Esto ha sido la perdición de muchos diagramas de Gantt hechos cuidadosamente.

Además, un ciclo de vida secuencial tiende a fomentar una relación de confrontación entre la gente que se va pasando el trabajo. “Me está pidiendo construir algo que no está en la especificación”. “Está cambiando de idea”. “No puedo hacerme responsable de algo que no controlo”. Y esto nos lleva a otra observación sobre el desarrollo secuencial – no es muy divertido. El modelo en cascada causa desdicha a la gente que hace productos. A los productos resultantes les falta mucha creatividad, habilidades y pasión de sus creadores. Las personas no son robots, y un proceso que les obligue a actuar como robots provoca infelicidad.

Un proceso rígido y con resistencia al cambio creará productos mediocres. Puede que los clientes obtengan lo que pidieron al principio (después de al menos dos pasos de traducción). Pero, ¿es eso de verdad lo que quieren una vez que ven el producto? Al tomar todos los requisitos al principio y dejarlos fijos, el producto está condenado a ser como mucho tan bueno como la idea inicial, en vez de ser lo mejor una vez que la gente ha aprendido o descubierto nuevas cosas.

Muchos seguidores de ciclos de vida secuencial experimentan estos problemas continuamente. Pero parece tan lógico que la reacción común suele ser pensar que el problema es suyo: “Si lo hiciéramos mejor funcionaría” – si hiciéramos más planificación, más documentación, ponemos más difícil los cambios, todo funcionaría bien. Por desgracia a muchos equipos les pasa justo lo contrario: Cuanto más lo intentan, peor. Además hay equipos de gestión que han apostado su reputación –y muchos recursos- por un modelo en cascada; cambiar a un modelo fundamentalmente diferente es, en apariencia, admitir que se ha cometido un error. Y Scrum es fundamentalmente diferente...

Desarrollo ágil y Scrum

La familia de métodos de desarrollo ágiles evolucionó a partir de los conocidos ciclos de vida incremental e iterativo. Nacieron de la creencia que un acercamiento más en contacto con la realidad humana –y la realidad del desarrollo de productos basados en el aprendizaje, innovación y cambio- daría mejores resultados. Los principios ágiles ponen el énfasis en construir software que funcione que se pueda usar rápidamente, en vez de pasarse mucho tiempo al principio escribiendo especificaciones. El desarrollo ágil se centra en equipos multifuncionales con capacidad para decidir por ellos mismos, en vez de grandes jerarquías y divisiones por funcionalidad. Y se centra en iteraciones rápidas, con el cliente dando su opinión continuamente. Suele pasar que cuando la gente oye hablar sobre desarrollo ágil o Scrum hay un gesto de reconocimiento – se parece mucho a lo que se hacía antes, cuando simplemente “lo hacíamos”.

El método ágil más popular es Scrum. Tuvo una fuerte influencia de un artículo de 1986 en el *Harvard Business Review* sobre prácticas asociadas con grupos exitosos de desarrollo de producto; en este artículo se introdujo el término “Scrum” (melé), relacionando el desarrollo exitoso de productos con el juego del *rugby* en el que un equipo auto-organizado (auto-gestionado) se mueve junto por el campo de desarrollo de productos. El primer equipo de Scrum lo creó Jeff Sutherland en Easel Corporation en 1993 y el *marco de trabajo* Scrum lo formalizó Ken Schwaber en 1995. Hoy en día Scrum es usado por empresas de todos los tamaños tales como Yahoo!, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne y la Reserva Federal de EE.UU. Muchos equipos que usan Scrum dicen haber obtenido mejoras sustanciales, y en algunos casos una completa transformación de la productividad y la moral. Para desarrolladores de producto –muchos de los cuales están quemados por los constantes cambios de tendencia en gestión- esto es significativo. Scrum es simple y poderoso.

Resumen de Scrum

Scrum es un *marco de trabajo* iterativo e incremental para el desarrollo de proyectos, productos y aplicaciones. Estructura el desarrollo en ciclos de trabajo llamados **Sprints**. Son iteraciones de 1 a 4 semanas, y se van sucediendo una detrás de otra. Los Sprints son de duración fija – terminan en una fecha específica aunque no se haya terminado el trabajo, y *nunca se alargan. Se limitan en tiempo*. Al comienzo de cada Sprint, un equipo multi-funcional selecciona los **elementos** (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no se pueden cambiar los elementos elegidos. Todos los días el equipo se reúne brevemente para informar del progreso, y actualizan unas gráficas sencillas que les orientan sobre el trabajo restante. Al final del Sprint, el equipo revisa el Sprint con los interesados en el proyecto, y les enseña lo que han construido. La gente obtiene comentarios y observaciones que se puede incorporar al siguiente Sprint. Scrum pone el énfasis en productos que funcionen al final del Sprint que realmente estén “hechos”; en el caso del software significa que el código esté integrado, completamente probado y potencialmente para entregar. Los roles, artefactos y eventos principales se resumen en la Figura 1.

Un tema importante en Scrum es “inspeccionar y adaptar”. El desarrollo inevitablemente implica aprender, innovación y sorpresas. Por eso Scrum hace hincapié en dar un pequeño paso de desarrollo; inspeccionar el producto resultante y la eficacia de las prácticas actuales; y entonces adaptar el objetivo del producto y las prácticas del proceso. Y volver a repetir.

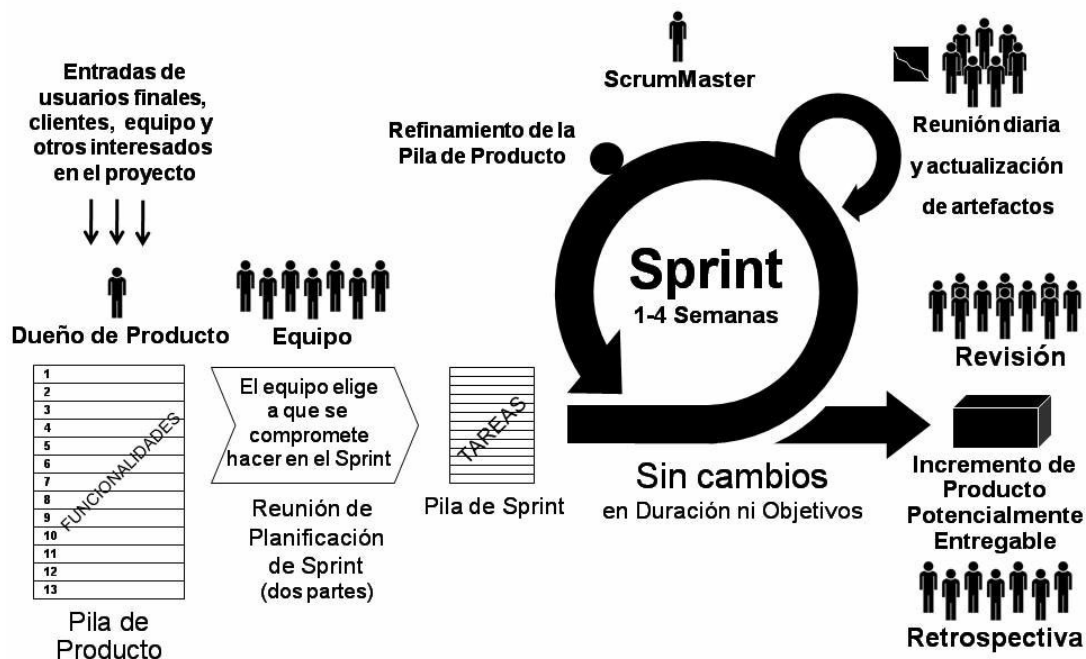


Figura 1. Scrum

Roles en Scrum

En Scrum hay 3 roles principales: El Dueño de Producto (DP), el Equipo y el ScrumMaster (SM). El **Dueño de Producto** es el responsable de maximizar el retorno de inversión (ROI) identificando las funcionalidades del producto, poniéndolas en una lista priorizada de funcionalidades, decidiendo cuales deberían ir al principio de la lista para el siguiente Sprint, y repriorizando y refinando continuamente la lista. El Dueño de Producto tiene la responsabilidad de las pérdidas y ganancias del producto, asumiendo que es un producto comercial. En el caso de una aplicación interna, el DP no es responsable del ROI en el mismo sentido de un producto comercial (que dará beneficio), pero es responsable de maximizar el ROI en el sentido de elegir - en cada Sprint - los elementos de más valor de negocio y menos coste. En algunas ocasiones el DP y el cliente son la misma persona; esto es muy común en aplicaciones internas. En otras, el cliente podría ser millones de personas con diferentes necesidades, en cuyo caso el rol de DP es parecido al rol de jefe de producto o jefe de marketing del producto que hay en muchas empresas. Sin embargo el Dueño de Producto es diferente al tradicional jefe de producto porque interactúa activa y frecuentemente con el equipo, estableciendo personalmente las prioridades y revisando el resultado en cada iteración - de 1 a 4 semanas-, en vez de delegar las decisiones de desarrollo en el jefe de proyecto. Es importante destacar que en Scrum hay una persona y sólo una, que hace - y tiene la autoridad final - de Dueño de Producto.

El **Equipo** construye el producto que va a usar el cliente, por ejemplo una aplicación o un sitio web. El equipo en Scrum es “multi-funcional” - tiene todas las competencias y habilidades necesarias para entregar un producto potencialmente distribuible en cada Sprint - y es “auto-organizado” (auto-gestionado), con un alto grado de autonomía y responsabilidad. En Scrum, los equipos se auto-organizan en vez de ser dirigidos por un jefe de equipo o jefe de proyecto. El equipo decide a que se compromete, y como hacer lo mejor para cumplir con lo comprometido; en el mundo de Scrum, al equipo se le conoce como “Cerdos” y a todos los demás como “Gallinas” (que viene de un chiste sobre un cerdo y una gallina que están hablando sobre abrir un restaurante llamado “Huevos con jamón”, y el cerdo no lo ve claro porque “él estaría verdaderamente comprometido, pero la gallina solo estaría implicada”).

El equipo en Scrum consta de siete personas más menos dos, y para un producto de software el equipo podría incluir analistas, desarrolladores, diseñadores de interface, y *testers*. El equipo desarrolla el producto y da ideas al DP de cómo hacer un gran producto. En Scrum, el equipo debería estar dedicado al 100% al trabajo en el producto durante el Sprint; intentando evitar hacer varias tareas en diferentes productos o proyectos. A los equipos estables se les asocia con una productividad más alta, así que evita cambiar miembros del equipo. A los grupos de desarrollo de aplicaciones con mucha gente se les organiza en varios equipos Scrum, cada uno centrado en diferentes funcionalidades del producto, coordinando sus esfuerzos muy de cerca. Dado que el equipo hace todo el trabajo (planificación, análisis, programación y pruebas) para una funcionalidad completa centrada en el cliente, a los equipos de Scrum también se les llama *equipos por funcionalidades*.

El **ScrumMaster** ayuda al grupo del producto a aprender y aplicar Scrum para conseguir valor de negocio. El ScrumMaster hace lo que sea necesario para ayudar a que el equipo tenga éxito. El ScrumMaster *no* es el jefe del equipo o jefe de proyecto; el ScrumMaster sirve al equipo, le protege de interferencias del exterior, y enseña y guía al DP y al equipo en el uso fructífero de Scrum. El ScrumMaster se asegura de que todo el mundo en el equipo (incluyendo al DP y la gerencia) entienda y siga las prácticas de Scrum, y ayuda a llevar a la organización, a través de los cambios necesarios y frecuentemente difíciles, a conseguir el éxito con el desarrollo ágil.

Como Scrum hace visibles muchos impedimentos y amenazas a la efectividad del DP y el equipo, es importante tener un ScrumMaster comprometido y que trabaje enérgicamente para ayudar a resolver dichos asuntos, o si no el equipo y el DP tendrán dificultades para tener éxito. Los equipos de Scrum deberían tener un ScrumMaster a tiempo completo, aunque en un equipo más pequeño podría ser un miembro del equipo (llevando una carga de trabajo más ligera). Un gran ScrumMaster puede venir de cualquier experiencia o disciplina previa: ingeniería, diseño, testing, gestión de productos, gestión de proyectos o gestión de calidad.

El ScrumMaster y el Dueño de Producto no pueden ser la misma persona; a veces el ScrumMaster necesitará parar los pies al DP (por ejemplo si intenta meter nuevas funcionalidades en mitad de un Sprint). Y al contrario de un jefe de proyecto, el ScrumMaster no le dice a gente las tareas que tienen asignadas – lo que hace es facilitar el proceso, apoyando al equipo que se organiza y gestiona solo. Si el ScrumMaster tuvo un puesto de gestión en el equipo, necesitará cambiar radicalmente su forma de pensar y el estilo de comunicación con el equipo para tener éxito con Scrum. En el caso de una transición de antiguo jefe a ScrumMaster, es mejor que esté en un equipo diferente al equipo en el que era el jefe, si no habrá un conflicto potencial por las dinámicas sociales y de poder.

Ten en cuenta que no existe el rol de jefe de proyecto en Scrum. A veces un (ex-)jefe de proyecto pasa a ser ScrumMaster, esto tiene un historial de éxito variado – hay una diferencia fundamental entre los dos roles en las responsabilidades del día a día y en la mentalidad necesaria para tener éxito. Una buena manera de comprender en profundidad el rol de ScrumMaster y empezar a desarrollar las habilidades necesarias es con los cursos de Certified ScrumMaster de la Scrum Alliance.

Además de estos tres roles, hay otros que contribuyen al éxito del producto, incluyendo los **jefes y gestores**. Aunque sus roles cambian en Scrum, siguen siendo valiosos. Por ejemplo:

- Ayudan al equipo respetando las reglas y el espíritu de Scrum
- Ayudan a quitar los impedimentos identificados por el equipo
- Ponen su experiencia y conocimiento a disposición del equipo

En Scrum, los jefes cambian el tiempo que dedicaban a hacer de “niñeras” (asignar tareas, pedir informes de estado y otras formas de micro-gestión) por tiempo como “gurus” o “sirvientes” del equipo (*mentoring*, *coaching*, ayudar a quitar obstáculos, ayudar a resolver problemas, dar ideas creativas y guiar el desarrollo de habilidades de los miembros del equipo). Para llevar a cabo este cambio los gestores pueden que necesiten cambiar su estilo de gestión; por ejemplo usar cuestionamiento socrático para ayudar al equipo a descubrir la solución a un problema en lugar de simplemente decidir una solución e imponérsela al equipo.

Comenzando con Scrum

El primer paso en Scrum consiste en que el Dueño de Producto articule la visión del producto. Al final esto evolucionará hacia una lista priorizada de funcionalidades llamada la **Pila de Producto**. Esta pila de producto existe (y evoluciona) a lo largo de la vida del proyecto; es el plan de trabajo del producto (Figura 2). En este punto, la Pila de Producto es la vista única y definitiva de “todo lo que podría ser hecho por el equipo en algún momento, en orden de prioridad”. Solo existe una única Pila de Producto; esto significa que el Dueño de Producto tiene que decidir sobre la priorización de todo el espectro.

Estimación de Esfuerzo Nueva Lo que queda del Sprint...										
Elemento	Detalles (URL wiki)	Prioridad	Estimación del Valor	Estimación de Esfuerzo Inicial	1	2	3	4	5	6
Como comprador, quiero poner un libro en el carrito de la compra (ver bocetos en el wiki)	...	1	7	5						
Como comprador, quiero quitar un libro del carrito de la compra	...	2	6	2						
Mejorar el rendimiento del procesador de transacciones (ver métricas de objetivos de rendimiento en el wiki)	...	3	6	13						
Investigar soluciones para acelerar la validación de tarjetas de crédito (ver métricas de objetivos de rendimiento en el wiki)	...	4	6	20						
Actualizar todos los servidores a Apache 2.2.3	...	5	5	13						
Diagnosticar y arreglar los errores de los scripts de procesamiento de órdenes (ID bugzilla 14823)	...	6	2	3						
Como comprador, quiero crear y guardar una lista de regalo	...	7	7	40						
Como comprador, quiero añadir o borrar elementos en mi lista de regalo	...	8	4	20						

Figura 2. La Pila de Producto

La Pila de Producto incluye una serie de **elementos**, principalmente funcionalidades nuevas del cliente (“permitir que todos los usuarios pongan libros en el carrito de la compra”), pero también objetivos de mejora de ingeniería (“rehacer el módulo de procesamiento de transacciones para hacerlo escalable”), trabajo de investigación o exploratorio (“investigar soluciones para acelerar la validación de tarjetas de crédito”), y posiblemente, fallos conocidos (“diagnosticar y arreglar los errores de script de procesamiento de órdenes”), si hay solo unos pocos fallos. (Un sistema con muchos errores normalmente tendrá además un sistema de seguimiento de incidencias). A muchos les gusta articular los requisitos usando “historias de usuario”, descripciones claras y concisas de la funcionalidad en términos del valor que aporta al usuario final del producto.

El subconjunto de la Pila de Producto que va en la entrega actual se conoce como la **Pila de Entrega**, y es en general el objetivo principal del Dueño de Producto.

La Pila de Producto es actualizada continuamente por el Dueño de Producto para reflejar los cambios de necesidad del cliente, nuevas ideas, movimientos de los competidores, dificultades técnicas, etc. El equipo da al Dueño de Producto las estimaciones del esfuerzo requerido para cada elemento de la Pila de Producto. Además, el Dueño de Producto es responsable de asignar una *estimación del valor de negocio* a cada elemento individual. Esto es normalmente una práctica desconocida para el Dueño de Producto. Por esa razón, a veces el ScrumMaster puede enseñar al Dueño de Producto a hacerlo. Con estas dos estimaciones (esfuerzo y valor) y quizás con estimaciones adicionales de riesgo, el Dueño de Producto prioriza la pila (realmente solo hace falta el subconjunto de la Pila de Entrega) para maximizar el ROI (eligiendo elementos de alto valor con poco esfuerzo) o secundariamente, para reducir algún riesgo importante. Como se verá más adelante, estas estimaciones del valor y esfuerzo se pueden actualizar en cada Sprint a medida que se aprenden cosas nuevas; por tanto, es una actividad continua de re-priorización de la Pila de Producto que evoluciona constantemente.

Scrum no dice como estimar la Pila de Producto, pero es común usar *estimaciones relativas* expresadas como “puntos” en vez de unidades absolutas de esfuerzo tales como personas-semana.

Con el tiempo un equipo calcula cuantos puntos relativos implementan por Sprint, por ejemplo, una media de 26 puntos por Sprint. Con esta información pueden proyectar una

fecha de entrega con todas las funcionalidades terminadas, o cuantas funcionalidades pueden ser completadas en una fecha determinada.

Los elementos de la Pila de Producto pueden variar significativamente en tamaño y esfuerzo. Los elementos grandes se parten en elementos más pequeños durante el taller de refinamiento de la Pila de Producto o la Reunión de Planificación del Sprint, y los elementos pequeños pueden ser consolidados.

Uno de los mitos sobre Scrum es que te impide escribir especificaciones detalladas; en realidad depende del Dueño de Producto y Equipo el decidir cuánto detalle es necesario, y esto variará de un elemento de la pila a otro, dependiente de las ideas del equipo y otros factores. Hay que poner lo que es importante en la menor cantidad de espacio necesario –en otras palabras, no describas todos los detalles posibles de un elemento, simplemente deja claro lo necesario para que se comprenda. Los elementos de baja prioridad y que están lejos de ser implementados tienen los requisitos menos detallados. Los elementos de alta prioridad que serán implementados próximamente tienden a tener más detalle.

Planificación del Sprint

Al comienzo de cada Sprint se hace la **Reunión de Planificación del Sprint**. Se divide en dos reuniones, la primera de las cuales se llama **Planificación del Sprint Parte Uno**.

En la Planificación del Sprint Parte Uno, el Dueño de Producto y Equipo (con la facilitación del ScrumMaster) revisan los elementos de alta prioridad de la Pila de Producto que el DP está interesado en implementar para este Sprint. Hablan sobre los objetivos y el contexto de dichos elementos de alta prioridad de la Pila de Producto, de tal forma que el Equipo se haga una idea de lo que piensa el Dueño de Producto. El DP y Equipo también revisan la “Definición de Hecho” que todos los elementos deben cumplir, tales como, “Hecho significa codificado según los estándares, revisado, implementado con desarrollo orientado a pruebas (TDD – *test-driven development*), probado 100% con pruebas automáticas, integrado y documentado”. La Parte Uno se centra en entender *qué* quiere el DP. Según las reglas de Scrum, al final de la Parte Uno el (siempre ocupado) DP se puede ir aunque *debe* estar disponible (por ejemplo, por teléfono) durante la siguiente reunión. Sin embargo, son bienvenidos a estar en la Parte Dos...

La Planificación del Sprint Parte Dos se centra en la planificación detallada de tareas para saber *cómo* implementar los elementos que el equipo decide hacer. El Equipo selecciona los elementos de la Pila de Producto a las que se comprometen que estará al final del Sprint, comenzando por la parte de arriba de la Pila de Producto (en otras palabras, comenzando con los elementos que tienen más prioridad para el DP) y escribiendo la lista en orden. Esto es una práctica clave en Scrum: El equipo decide a cuanto trabajo se compromete en vez de serles asignado por el Dueño de Producto. Esto hace que el compromiso sea más fiable porque el equipo lo está haciendo basado en su propio análisis y planificación en vez de que venga “hecho” por otros. Aunque el DP no tiene control sobre la cantidad de trabajo que el equipo se compromete a hacer, sabe que los elementos que el equipo coja son de lo alto de la Pila de Producto –en otras palabras, los elementos que ha clasificado como los más importantes. El equipo también tiene autoridad para seleccionar elementos de más abajo de la lista, esto normalmente pasa cuando el equipo y Dueño de Producto observan que algunos elementos de más baja prioridad encajan fácil y apropiadamente entre los de alta prioridad.

La Reunión de Planificación del Sprint normalmente durará unas cuantas horas –el equipo se está comprometiendo seriamente a completar el trabajo, y este compromiso necesita ser meditado para poder tener éxito. El equipo probablemente empezará la Planificación del

Sprint Parte Dos estimando cuánto tiempo tiene cada miembro para trabajo relacionado con el Sprint – en otras palabras, su día laboral medio menos el tiempo que pasarán en reuniones, leyendo el correo, comiendo, etc. Para la mayoría de la gente esto significa 4-6 horas de tiempo disponible al día para trabajo relacionado con el Sprint.

Ver figura 3.

Longitud del Sprint	2 semanas
Días laborables durante el Sprint	8 días

Miembro del equipo	Días disponibles durante el Sprint*	Horas disponibles por día	Total horas disponibles
Tracy	8	4	32
Sanjay	7	5	35
Phillip	8	4	32
Jing	6	5	30

* Neto de vacaciones y otros días fuera de la oficina

Figura 3. Estimando horas disponibles

Después de haber determinado el tiempo disponible el equipo empieza con el primer elemento de la Pila de Producto –en otras palabras, el elemento de más alta prioridad del DP- y entre todos los dividen en tareas individuales, que se guardan en un documento llamado **Pila del Sprint** (Figura 4). Como se ha mencionado anteriormente, el DP debe estar disponible durante la Parte Dos (por ejemplo vía telefónica) para cualquier clarificación necesaria. El equipo se irá moviendo secuencialmente hacia abajo de la Pila de Producto de esta forma hasta que se acaben las horas disponibles. Al final de la reunión el equipo habrá producido una lista de tareas con estimaciones (típicamente en horas o fracciones de un día).

Scrum anima a tener trabajadores multi-funcionales en vez de simplemente “trabajar según el nombre del puesto de trabajo” tal como “tester” que solo prueba. En otras palabras, los miembros del equipo “van donde está el trabajo” y ayudan en todo lo posible. Si hay muchas tareas de pruebas, entonces *todo* el equipo puede ayudar. Esto no significa que todos son generalistas; por supuesto que hay personas entrenadas especialmente en probar (y lo mismo con otras cosas) pero los miembros del equipo trabajan conjuntamente y aprenden nuevas habilidades de los demás. Por lo tanto durante la generación de tareas y estimación de la Planificación del Sprint, no es necesario –ni apropiado- que la gente se ofrezca voluntario para las tareas que “pueden hacer mejor”. En vez de eso, es mejor ofrecerse voluntario para una sola tarea a la vez, y cuando hay que coger otra tarea considerar tareas cuyo propósito incluya aprender (quizá haciendo programación en parejas con un especialista).

Dicho eso, hay algunas ocasiones *excepcionales* en las que *Juan* puede hacer una tarea particular porque a otros miembros del equipo les llevaría mucho o tiempo o sería imposible que lo aprendieran–quizás Juan es la única persona con cualidades artísticas para dibujar. Otros miembros del equipo no podrían dibujar un “muñecote” incluso si su vida dependiera de ello. En este caso excepcional – y si no es excepcional y no se va haciendo infrecuente a medida que el equipo aprende, entonces algo va mal- puede ser necesario preguntarse si las tareas de dibujo planeadas que Juan *debe* hacer son posibles dentro de un Sprint tan corto.

Muchos equipos también usan una herramienta visual de seguimiento de tareas, en forma de un gran tablero de tareas en la pared donde las tareas (escritas en post-its) se cambian durante el Sprint entre columnas etiquetadas con “No empezado”, “En progreso” y “Completado”. Ver Figura 5.

Elemento de la Pila de Producto	Tarea del Sprint	Voluntario	Esfuerzo estimado inicial	Nuevo esfuerzo estimado Lo que queda al final del día...					
				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 (interfaz de usuario)		8						
	crear página web (lógica Javascript)		13						
	escribir pruebas de aceptación automáticas		13						
	actualizar la página de ayuda del comprador		3						
	...								
Mejorar el rendimiento de procesamiento de transacciones	juntar el código DCP y completar los test del nivel de capa		5						
	completar la orden de máquina para pRank		8						
	Cambiar el DCP y el lector para usar el API http de pRank		13						

Figura 4. Pila de Producto



Figura 5. Gestión visual – Tareas de la Pila de Sprint en la pared

Uno de los pilares de Scrum es que una vez que el Equipo se compromete, cualquier cambio o adición debe esperar al siguiente Sprint. Esto significa que si a mitad del Sprint el Dueño de Producto decide que hay un nuevo elemento en el que debería trabajar el equipo, no puede hacer el cambio hasta el comienzo del siguiente Sprint. Si aparece alguna circunstancia externa que haga cambiar las prioridades significativamente, y significa que el equipo estaría perdiendo el tiempo si continua con el trabajo actual, el Dueño de Producto o el equipo puede terminar el Sprint. El equipo para, y una nueva reunión de Planificación del Sprint da comienzo a un nuevo Sprint. El trastorno de hacer esto suele ser grande; esto sirve para desincentivar que el Dueño de Producto o el equipo recurran frecuentemente a esta dramática decisión.

Existe una influencia positiva y poderosa en el hecho de que el equipo esté protegido de cambios en los objetivos durante el Sprint. Primero, el equipo va a trabajar con la certeza absoluta de que sus compromisos no cambiarán, eso refuerza que el equipo se centre en asegurarse que terminan. Segundo, disciplina al Dueño de Producto para que piense seriamente en como prioriza los elementos de la Pila de Producto que da al equipo para el Sprint.

El Dueño de Producto gana dos cosas siguiendo estas reglas de Scrum. Primero, tiene la confianza de saber que el equipo se ha comprometido a completar un conjunto de trabajo claro y realista que ellos han elegido. Con el tiempo puede convertirse en un equipo muy experto en la elección y el cumplimiento de un compromiso realista. Segundo, el Dueño de Producto puede hacer cualquier cambio que quiera a la Pila de Producto antes de que empiece el *siguiente* Sprint. En ese momento, adiciones, supresiones, modificaciones y repriorizaciones son posibles y aceptables. Aunque el Dueño de Producto no puede hacer cambios a los elementos seleccionados durante el Sprint actual, está solo a un Sprint de duración o menos de hacer los cambios que desee. Se acabó el estigma en torno al cambio – cambio de dirección, cambio de requisitos, o simplemente cambio de opinión- y puede ser por esta razón que los Dueños de Producto están normalmente tan entusiasmados por Scrum como cualquier otro.

Scrum Diario

Una vez empezado el Sprint, el equipo participa en otra de las prácticas claves de Scrum: El **Scrum Diario**. Es una reunión corta (15 minutos o menos) que se celebra todos los días a una hora prefijada. Todo el equipo asiste a la reunión. Para hacerla corta, se recomienda que todos estén de pie. Es la oportunidad del equipo de informar a los demás sobre el progreso y los obstáculos. En el Scrum Diario, cada miembro del equipo, uno por uno, informa sobre tres (y solo tres) cosas *a los otros miembros del equipo*: (1) Que han hecho desde la última reunión; (2) Que tienen planificado hacer antes de la siguiente reunión; y (3) cualquier bloqueo o impedimento que tengan. Hay que tener en cuenta que el Scrum Diario no es una reunión de estado para informar a un jefe; es tiempo que dedica un equipo auto-organizado para compartir entre sus miembros lo que está pasando, para ayudarles a coordinarse. Alguien anota los bloqueos, y el ScrumMaster se responsabiliza de ayudar a los miembros del equipo a resolverlos. No hay discusiones durante el Scrum Diario, solo se da respuesta a las tres preguntas; si se necesita hablar sobre algo se hace después del Scrum Diario en una reunión de seguimiento, aunque Scrum no dicta que sea obligatorio ir a dicha reunión. Esta reunión de seguimiento es algo común cuando el equipo se adapta a la información que oyeron en el Scrum Diario: en otras palabras, otro ciclo de inspección y adaptación. Normalmente se aconseja que *no* acudan al Scrum Diario ni jefes, ni gerentes ni otros puestos de autoridad percibida. Esto puede hacer sentir al equipo que está siendo “observado” –con la presión de informar de un gran progreso todos los días (una expectativa poco realista), y cohibe el informar sobre problemas –y tiende a socavar la autogestión del equipo, e invita a la micro-gestión. Es más útil que los interesados en el proyecto se acerquen al equipo después de la reunión y se ofrezcan a ayudar con cualquier bloqueo que está haciendo disminuir el progreso del equipo.

Actualizando la Pila de Sprint y la Gráfica de Trabajo Restante del Sprint

Todos los días el equipo actualiza sus estimaciones de la cantidad de trabajo que queda para terminar sus tareas actuales en la **Pila de Sprint** (Figura 6). Después de esta actualización, alguien suma las horas restantes del equipo como un todo, y dibuja una **Gráfica de Trabajo Restante del Sprint** (Figura 7). Esta gráfica muestra cada día una nueva estimación de cuanto trabajo queda (medido en personas-hora) para terminar las tareas del equipo. Idealmente es una gráfica de pendiente *descendiente* que está en la trayectoria para llegar a “no queda nada de esfuerzo” el último día del Sprint. Y aunque a veces tiene buena pinta, frecuentemente no la tiene; es la realidad del desarrollo de productos. Lo importante es que muestre al equipo el

progreso hacia su objetivo, no en términos de cuanto *bicimos* en el pasado (un hecho irrelevante en término de *progreso*), sino en término de cuanto trabajo *queda en el futuro* –lo que separa al equipo de su objetivo. Si la línea de trabajo restante no está bajando hasta la finalización del trabajo cuando se acerca el final del Sprint, entonces el equipo necesita hacer ajustes, por ejemplo reducir el alcance del trabajo o encontrar una forma de trabajar más eficientemente al tiempo que mantienen un ritmo sostenible.

Aunque se puede usar una hoja de cálculo para crear y mostrar la Gráfica de Trabajo Restante, muchos equipos encuentran más efectivo tenerlo en papel en una pared, con actualizaciones en rotulador; esta solución de “poca tecnología y mucho contacto” es rápida, simple y frecuentemente más visible que una gráfica de ordenador.

				Nuevo esfuerzo estimado Lo que queda al final del día...					
Elemento de la Pila de Producto	Tarea del Sprint	Voluntario	Esfuerzo estimado inicial	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 (interfaz de usuario)	Jing	3	3	3	2	0	0	
	crear página web (lógica Javascript)	Tracy & Sam	2	2	2	2	1	0	
	escribir pruebas de aceptación automáticas	Sarah	5	5	5	5	5	0	
	actualizar la página de ayuda del comprador	Sanjay & Jing	3	3	3	3	3	0	
	...								
Mejorar el rendimiento de procesamiento de transacciones	juntar el código DCP y completar los test del nivel de capa		5	5	5	5	5	5	
	completar la orden de máquina para pRank		3	3	8	8	8	8	
	Cambiar el DCP y el lector para usar el API http de pRank		5	5	5	5	5	5	
Total (personas-hora)			...	50	49	48	44	43	34

Figura 6. Actualizaciones diarias de Trabajo Restante en la Pila del Sprint

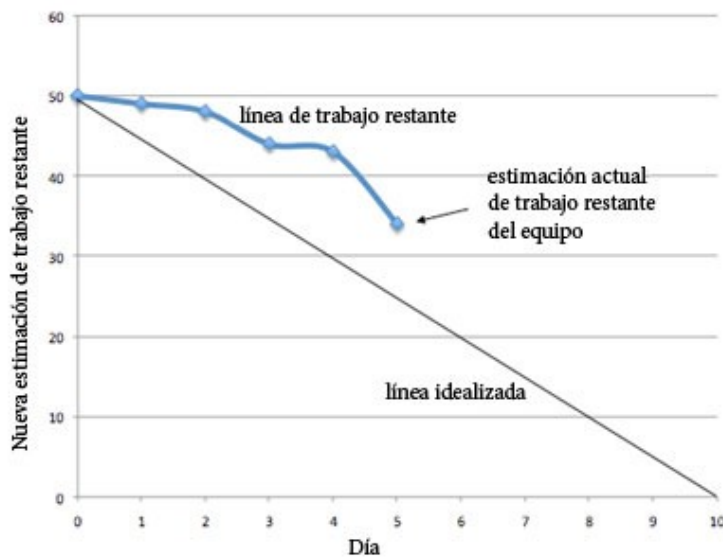


Figura 7. Gráfica de Trabajo Restante del Sprint

Refinamiento de la Pila del Producto

Una de las directrices menos conocidas pero más útiles de Scrum es que el equipo debe dedicar un cinco o diez por ciento de cada Sprint a refinar o “asear” la Pila de Producto. Esto incluye análisis detallado de requisitos, separar elementos grandes en otros más pequeños, estimación de nuevos elementos, y reestimación de elementos existentes. Scrum no dice nada sobre cómo hay que hacerlo, pero nosotros sugerimos hacer un taller centrado en esto cerca del final del Sprint, para que el equipo y Dueño de Producto se puedan dedicar a este trabajo sin interrupciones. Para un Sprint de dos semanas, cinco por ciento de duración significa que en cada Sprint hay medio día de taller de Refinamiento de la Pila de Producto. Esta actividad de refinamiento no se hace sobre los elementos seleccionados para el Sprint actual; es para los elementos del futuro, probablemente para el siguiente o el de dentro de dos Sprints. Con esta práctica, la Planificación del Sprint se hace relativamente más sencilla porque el Dueño de Producto y el Equipo empiezan la planificación con un conjunto de elementos claro, bien analizado y cuidadosamente estimado. Una señal de que este taller de refinamiento no se está haciendo (o se está haciendo mal) es que la Planificación del Sprint tenga muchas preguntas, muchos nuevos descubrimientos o mucha confusión.

Terminando el Sprint

Uno de los principios fundamentales de Scrum es que nunca se prolonga la duración del Sprint –termina en la fecha asignada aunque el equipo no haya terminado el trabajo comprometido. Los equipos normalmente se comprometen en exceso en su primer Sprint y no cumplen sus objetivos. Podrían entonces compensarlo al alta y comprometerse por debajo, y terminar antes. Alrededor del tercer o cuarto Sprint, los equipos suelen haber averiguado de que son capaces de entregar (la mayor parte del tiempo), y cumplirán los objetivos del Sprint con más fiabilidad. A los equipos se les recomienda que escojan una duración para el Sprint (por ejemplo dos semanas) y que no la cambien. Una duración consistente ayuda al equipo a saber cuánto pueden hacer, que les ayuda en la estimación y en la planificación de la entrega a más largo plazo. También ayuda a que el equipo coja un ritmo de trabajo; en Scrum a esto se le llama el “pulso” del equipo.

Revisión del Sprint

Cuando termina el Sprint se hace la **Revisión del Sprint**, donde el equipo revisa el Sprint junto con el Dueño de Producto. A menudo se le llama incorrectamente la “demo” pero esto no capta la intención real de esta reunión. Una idea clave en Scrum es *inspeccionar y adaptar*. Ver y aprender lo que está pasando y entonces evolucionar basándose en los comentarios y observaciones, en ciclos repetidos. La Revisión del Sprint es una actividad de inspección y adaptación del *producto*. Es la oportunidad de que el Dueño de Producto vea lo que está pasando con el producto y con el equipo (o sea, revisión del Sprint); y la oportunidad del equipo de saber cómo va el Dueño de Producto y el mercado. Por lo tanto el elemento más importante de la Revisión es una *conversación* en profundidad entre el equipo y el DP para conocer la situación, recibir consejo, etc. La revisión incluye una demo de lo que el equipo construyó durante el Sprint, pero si el centro de la revisión es la demo en vez de la conversación, estará desbalanceado.

Una directriz útil –aunque frecuentemente pasada por alto– es que el ScrumMaster es responsable de saber la “Definición de Hecho” que fue definida durante la Planificación del

Sprint, y durante esta reunión es responsable de decir al Dueño de Producto si alguno de los elementos implementados por el equipo no cumple esta definición. De esta forma, hay una mayor visibilidad de la calidad del trabajo; los equipos no pueden simular la calidad presentando software que parece que funciona bien, pero que está implementado con código desordenado, sin probar y de mala calidad.

En esta reunión están presentes el DP, los miembros del Equipo, y el ScrumMaster, además de los clientes, gente interesada en el proyecto, expertos, directivos y cualquier otra persona que lo desee. La parte de demo de la Revisión del Sprint no es una “presentación” que da el equipo – no hay diapositivas. Una guía de Scrum es que no se debería dedicar más de 30 minutos a preparar la demo, si no quiere decir que algo va mal en el trabajo del equipo. Es simplemente una demo de lo que se ha construido. Cualquiera es libre de preguntar y dar su opinión.

Retrospectiva del Sprint

La Revisión del Sprint implica inspeccionar y adaptar el *producto*. La **Retrospectiva del Sprint**, que va después de la Revisión, implica inspeccionar y adaptar el *proceso*. Es una práctica que se saltan algunos equipos y es una pena porque es el mecanismo principal para obtener la visibilidad que Scrum proporciona en áreas de mejora potencial, y transformarlo en resultados. Es una oportunidad para que el equipo hable sobre lo que funciona y lo que no, y acuerden que cambios quieren intentar. El Equipo y ScrumMaster tienen que asistir, y el Dueño de Producto es bienvenido pero no hace falta que acuda. A veces el ScrumMaster puede actuar como un facilitador efectivo para la retrospectiva, pero puede ser mejor encontrar a alguien neutral de fuera para facilitar la reunión; una buena forma de abordar esto es que los ScrumMasters faciliten las retrospectivas de otros equipos, que además ayuda a la polinización cruzada entre equipos.

Una forma simple de estructurar la Retrospectiva del Sprint es dibujar dos columnas en una pizarra con los textos “Qué fue bien” y “Qué se podría mejorar” –y entonces que la gente se mueva por la sala añadiendo elementos a las listas. Cuando algún elemento se repite, se le añade una marca para dejar claro cuáles son los elementos comunes. Entonces el equipo busca las causas subyacentes, acuerda aplicar un número pequeño de cambios en el siguiente Sprint, y se compromete a revisar los resultados en la Retrospectiva del próximo Sprint.

Una práctica útil al final de la Retrospectiva es que el equipo etiquete los elementos de ambas columnas con una “C” si es *causado* por Scrum (en otras palabras, sin Scrum no hubiera sucedido), una “E” si ha sido *expuesto* por Scrum (en otras palabras, hubiera sucedido con o sin Scrum, pero Scrum lo da a conocer al equipo), o una “U” si *no está relacionado* con Scrum (como el tiempo atmosférico). El equipo puede encontrar muchas Cs en el lado de “Qué fue bien”, y un montón de Es en “Que se podría mejorar”; esto son buenas noticias incluso aunque la lista de “Qué se podría mejorar” sea larga porque el primer paso para resolver los asuntos subyacentes es hacerlos visibles, y Scrum es un potente catalizador para ello.

Actualizando la Pila de Entrega y la Gráfica de Trabajo Restante

Llegados a este punto se han terminado algunos elementos, algunos se han añadido, algunos tienen nuevas estimaciones, y algunos se han caído del objetivo de entrega. El Dueño de Producto es responsable de asegurarse que estos cambios se reflejan en la Pila de Entrega (y más ampliamente, en la Pila de Producto). Además, Scrum incluye una gráfica de **Trabajo**

Restante de la Entrega que muestra el progreso hecho hacia la fecha de entrega. Es análogo a la gráfica de Trabajo Restante del Sprint, pero a un nivel más alto (requisitos) en vez de tareas de grano fino. Como es poco probable que un nuevo Dueño de Producto sepa por qué o cómo crear esta gráfica, es otra buena oportunidad para que el ScrumMaster ayude al Dueño de Producto. Ver Figura 8 y Figura 9 para un ejemplo de la Pila de Entrega y la gráfica de Trabajo Restante de la Entrega.

Estimación de Esfuerzo Nueva Lo que queda del Sprint...										
Item	Detalles (URL wiki)	Prioridad	Estimación del Valor	Estimación de Esfuerzo Inicial	1	2	3	4	5	6
Como comprador, quiero poner un libro en el carrito de la compra (ver bocetos en el wiki)	...	1	7	5	0	0	0			
Como comprador, quiero quitar un libro del carrito de la compra	...	2	6	2	0	0	0			
Mejorar el rendimiento del procesador de transacciones (ver métricas de objetivos de rendimiento en el wiki)	...	3	6	13	13	0	0			
Investigar soluciones para acelerar la validación de tarjetas de crédito (ver métricas de objetivos de rendimiento en el wiki)	...	4	6	20	20	20	0			
Actualizar todos los servidores a Apache 2.2.3	...	5	5	13	13	13	13			
Diagnosticar y arreglar los errores de los scripts de procesamiento de órdenes (ID bugzilla 14823)	...	6	2	3	3	3	3			
Como comprador, quiero crear y guardar una lista de regalo	...	7	7	40	40	40	40			
Como comprador, quiero añadir o borrar elementos en mi lista de regalo	...	8	4	20	20	20	20			
...										
Total				537	580	570	500			

Figura 8. Pila de Entrega (subconjunto de la Pila de Producto)

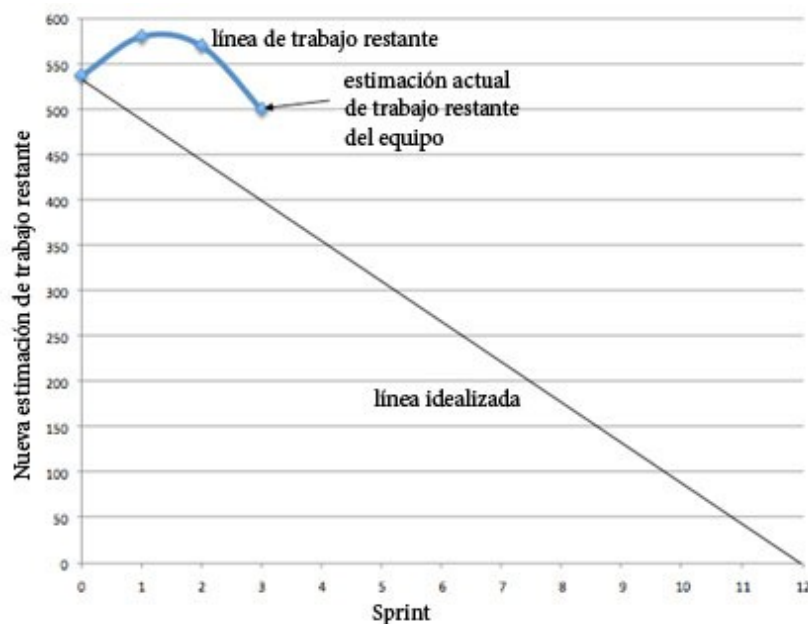


Figura 9. Gráfica de Trabajo Restante de la Entrega

Comenzando el próximo Sprint

Después de la Revisión del Sprint, el Dueño de Producto puede actualizar la Pila de Producto con nuevas ideas. En este punto, el Dueño de Producto y el equipo están listos para empezar otro ciclo de Sprint. No hay tiempo de descanso entre Sprints –los equipos normalmente van de la Retrospectiva del Sprint una tarde a la Planificación del próximo Sprint la mañana siguiente (o después del fin de semana).

Uno de los principios de desarrollo ágil es “ritmo sostenible”, y solo trabajando dentro de las horas delimitadas por el horario laboral a un ritmo razonable permite que los equipos continúen este ciclo indefinidamente.

Sprint de Entrega

La visión de perfección de Scrum es que el producto es potencialmente entregable al final de cada Sprint, lo que implica que no se necesitan trabajos de acabado tales como pruebas o documentación. Por contra, implica que todo está completamente terminado en cada Sprint; y se podría realmente empaquetar o desplegar inmediatamente después de la Revisión del Sprint.

Sin embargo, muchas organizaciones tienen prácticas de desarrollo débiles y no pueden alcanzar esta visión de perfección, o hay otras circunstancias atenuantes (por ejemplo que “la máquina se estropeó”). En este caso, quedará algo de trabajo por hacer, por ejemplo las últimas pruebas de integración del entorno producción, y en este caso se necesitará un “Sprint de Entrega” para terminar este trabajo pendiente.

Hay que tener en cuenta que la necesidad de un Sprint de Entrega es una señal de alguna debilidad; lo ideal es que no sea necesario. Si fuera necesario, los Sprints continúan hasta que el Dueño de Producto decide que el producto está casi listo para entregar, en este punto habrá un Sprint de Entrega para tenerlo listo para el lanzamiento. Si el equipo ha seguido buenas prácticas de desarrollo, con refactorización e integración continua, y pruebas efectivas durante cada Sprint, debería ser necesario poco trabajo de estabilización y acabado antes de la entrega.

Planificación de la entrega y Refinamiento de la Pila de Producto inicial

Una pregunta que la gente se hace a veces es cómo en un modelo iterativo se puede hacer planificaciones de entregas a largo plazo. Hay dos casos a considerar: (1) un producto nuevo en su primera entrega, y (2) un producto ya existente en una entrega posterior.

En el caso de un producto nuevo, o *un producto existente que acaba de adoptar Scrum*, hay que refinar la Pila de Producto inicial antes del primer Sprint, el Dueño de Producto y el equipo dan forma a la Pila de Producto de Scrum. Podría llevar unos días o una semana, e implica un taller sobre la visión del producto, algo de análisis detallado de requisitos, y la estimación de todos los elementos identificados para la primera entrega.

Es sorprendente en Scrum que en el caso de un producto establecido con una Pila de Producto establecida no se debería necesitar ninguna planificación especial o extensiva para la siguiente entrega. ¿Por qué? Porque el Dueño de Producto y el equipo debería estar refinando la Pila de Producto en cada Sprint (cinco o diez por ciento del tiempo de cada Sprint), preparándose continuamente para el futuro. Este modo de *desarrollo de producto continuo* evita la

necesidad de las etapas dramáticas de preparar-ejecutar-concluir que se ven en desarrollos tradicionales de ciclo de vida secuencial.

Durante el taller de refinamiento inicial de la Pila de Producto y durante los refinamientos continuos de la pila de cada Sprint, el Equipo y el Dueño de Producto planificarán la entrega, perfeccionarán las estimaciones, prioridades y contenido a medida que van aprendiendo.

Algunas entregas tienen una fecha fijada; por ejemplo: “Entregaremos la versión 2.0 de nuestro proyecto en la feria del 10 de noviembre”. En esta situación, el equipo completará todos los Sprints posibles (con todas las funcionalidades posibles) en el tiempo disponible. Otros productos necesitan tener ciertas funcionalidades antes de estar terminados y el producto no se podrá lanzar hasta que cumpla dichos requisitos, independientemente del tiempo que lleve. Como Scrum enfatiza el producir código potencialmente entregable en cada Sprint, el Dueño de Producto puede optar por empezar a hacer entregas internas, para permitir que el cliente pueda beneficiarse del trabajo completado lo antes posible.

Como no es posible saber todo de antemano, la atención se centra en crear y perfeccionar un plan para darle una dirección amplia a la entrega, y clarificar como se tomarán las decisiones (por ejemplo alcance frente a fechas de entrega). Piensa en esto como si fuera la guía que te lleva hasta el destino final; los caminos exactos que tomarás y las decisiones que harás durante el viaje se pueden llevar a cabo durante el trayecto.

La mayoría de DP eligen el enfoque de una entrega. Por ejemplo, elegirán una fecha de entrega y trabajarán con el equipo para estimar los elementos de la Pila de Entrega que se podrán tener para esa fecha. En situaciones de compromisos de “precio fijo/fecha fija/funcionalidad fija”, al menos uno de estos parámetros tiene que tener un margen para permitir incertidumbre y cambio; en esto Scrum no difiere de otros enfoques.

Foco en la aplicación o el producto

Tanto en aplicaciones como en productos –para el mercado o para uso interno de la organización– Scrum mueve a los equipos del modelo *centrado en el proyecto* hacia un modelo de *desarrollo continuo de aplicación/producto*. Ya no hay un proyecto con un principio, medio y fin. Y por tanto no hay un jefe de proyecto tradicional. En vez de eso hay simplemente un Dueño de Producto estable y un equipo auto-gestionado de larga duración que colaboran en series “infinitas” de Sprint de 1 a 4 semanas, hasta que se retira el producto o aplicación. El equipo y el dueño del negocio –que puede ser un cliente interno o del departamento de gestión de producto) hacen toda la gestión del “proyecto”. No lo gestiona ningún gestor de tecnología o alguien de la Oficina de Gestión de Proyectos.

Scrum también se puede usar en proyectos que son iniciativas únicas (en contraposición a trabajos para crear o evolucionar aplicaciones de larga duración); también en este caso la gestión del proyecto la hace el equipo y el Dueño de Producto.

¿Qué pasa si no hay trabajo nuevo suficiente para una o más de las aplicaciones existentes para garantizar un equipo de larga duración para cada aplicación? En este caso, un equipo estable de larga duración puede coger elementos de una aplicación en un Sprint, luego elementos de otra aplicación para el siguiente Sprint; en esta situación los Sprints suelen ser bastante cortos, por ejemplo de una semana.

Ocasionalmente puede haber poco trabajo incluso para la situación anterior, en ese caso el equipo puede coger elementos de *varias* aplicaciones durante el mismo Sprint; sin embargo hay que tener cuidado con esta solución porque puede llevar a cambios de tarea improductivos

sobre las diferentes aplicaciones. Un tema básico para conseguir productividad en Scrum es que el equipo tiene que estar *centrado* en un producto o aplicación durante un Sprint.

Retos habituales

Scrum no es solamente un conjunto concreto de prácticas – más bien y lo que es más importante, es un marco de trabajo que proporciona visibilidad al equipo y un mecanismo que les permite “inspeccionar y adaptar” en consecuencia. Scrum hace visible los impedimentos y disfunciones que están impactando en la efectividad del Dueño de Producto y del equipo, a fin que puedan ser abordados. Por ejemplo, el DP puede no conocer realmente el mercado, las funcionalidades, o como estimar el valor de negocio relativo. O el equipo puede no tener las habilidades necesarias para el trabajo de desarrollo o para la estimación del esfuerzo.

El marco de trabajo Scrum revelará estas debilidades rápidamente. Scrum no soluciona los problemas de desarrollo; los hace dolorosamente visibles, y proporciona un marco de trabajo para que la gente explore nuevas formas de resolver los problemas en ciclos cortos y con pequeños experimentos de mejora.

Imagina que el equipo no entrega lo que había comprometido en el primer Sprint por un mal análisis y estimación de las tareas. Esto parece un fallo a ojos del equipo. Pero en realidad esta experiencia es el primer paso necesario para ser más realistas y reflexivos en los compromisos. Este patrón –de que Scrum ayude a hacer visibles las disfunciones, permitiendo que el equipo haga algo al respecto- es el mecanismo básico que produce los mayores beneficios experimentados por los equipos que usan Scrum.

Un fallo típico de los equipos cuando se encuentran con una práctica de Scrum que les resulta difícil es cambiar Scrum en vez de cambiar ellos. Por ejemplo, los equipos con problemas en la entrega de sus compromisos podrían decidir ampliar la duración del Sprint, así siempre tendrán tiempo –y en el proceso se aseguran de no tener que mejorar la estimación y gestión de su tiempo. De esta forma, sin un *coaching* y apoyo de un ScrumMaster experimentado las organizaciones pueden hacer que Scrum sea simplemente un reflejo de sus propias debilidades y disfunciones, y socavar los beneficios reales que ofrece Scrum: Hacer visible lo bueno y lo malo, y dar a la organización la oportunidad de moverse a un nivel más alto.

Otro fallo típico es asumir que una práctica está desaconsejada o prohibida simplemente porque Scrum no la dicta específicamente. Por ejemplo, Scrum no dice que el Dueño de Producto tenga una estrategia a largo plazo sobre el producto; tampoco dice que los ingenieros busquen consejo en los ingenieros más experimentados sobre problemas técnicos complejos. Scrum deja que los individuos involucrados tomen la decisión correcta; y en la mayoría de los casos son aconsejables estas prácticas (junto con muchas otras).

Otra cosa con la que hay tener cuidado es que los gestores o jefes impongan Scrum en sus equipos; Scrum consiste en dar al equipo espacio y herramientas para que se gestionen ellos mismos, y que se les obligue a esto desde arriba no es una receta para el éxito. Un mejor enfoque sería que el equipo aprendiera Scrum de un compañero o un jefe, que obtenga una educación completa con entrenamiento profesional, y después se tome la decisión como equipo de seguir las prácticas fielmente durante un periodo definido; al final del periodo, el equipo evaluará su experiencia y decidirá si quiere continuar.

Las buenas noticias son que aunque el primer Sprint suele ser muy difícil para el equipo, los beneficios de Scrum tienden a ser visibles al final del Sprint, haciendo que muchos equipos de Scrum exclamen: “Scrum es duro, ¡pero es mucho mejor que lo que estábamos haciendo antes!”.

Resultados con Scrum

Los beneficios sobre los que han informado los equipos incluyen diversos aspectos. En Yahoo! migramos casi 200 equipos a Scrum durante un periodo de tres años, con un total de 2000 personas. Estos proyectos iban desde sitios web con mucho diseño y enfocados al cliente tales como Yahoo! Fotos hasta la infraestructura de servicios de back-end de misión crítica como Yahoo Mail que da servicio a millones de clientes.

Varias veces al año hacemos una encuesta a todos los que usan Scrum en Yahoo (incluyendo los Dueños de Producto, Miembro del Equipo, ScrumMasters, y sus jefes funcionales) y les pedimos que comparen Scrum con el enfoque que usaban anteriormente. Este es el resumen de algunos de los datos obtenidos:

- **Productividad:** 68% respondieron que Scrum es mejor o mucho mejor (4 o 5 en una escala de 5); 5% dijeron que era peor o mucho peor (1 o 2 en una escala de 5); 27% dijeron que era lo mismo (3 en una escala de 5).
- **Moral del Equipo:** 52% dijeron que es mejor o mucho mejor; 9% dijeron que es peor o mucho peor; 39% dijeron que es lo mismo.
- **Adaptabilidad:** 63% dijeron que es mejor o mucho mejor; 4% dijeron que es peor o mucho peor; 33% dijeron que es lo mismo.
- **Responsabilidad:** 62% dijeron que es mejor o mucho mejor; 6% dijeron que es peor o mucho peor; 32% dijeron que es lo mismo.
- **Colaboración y Cooperación:** 81% dijeron que es mejor o mucho mejor; 1% dijeron que es peor o mucho peor; 18% dijeron que es lo mismo.
- **La productividad de los equipos mejoró una media del 36% según los DP.**
- **El 85% de los miembros de equipos dijeron que continuarían usando Scrum si la decisión dependiera exclusivamente de ellos (15% dijeron “No” o “No saben”).**