

Módulo 1. ¿Qué es Scrum?

Fundamentos y marco metodológico

La agilidad organizacional plantea un mundo rico y diverso donde, bajo ciertos lineamientos, conceptos y herramientas, podemos desarrollar o crear nuevas e innovadoras formas de generar productos, *softwares* o, en resumidas cuentas, entregables que aporten valor a las necesidades conocidas de un cliente.

Existen varios marcos de trabajo o *frameworks* en el mundo ágil, y todos apuntan o están orientados a la misma idea que se expresó anteriormente. Esta lectura nos convoca analizar uno de ellos, que resulta ser uno de los *frameworks* más utilizados dentro de un contexto de actuación dado.

Hablamos del marco Scrum, que toma su nombre de la formación deportiva utilizada en rugby. La analogía parte de la base de que en el rugby, el movimiento Scrum implica que el equipo genere sinergia y trabaje en conjunto para lograr un objetivo. La formación se entrelaza, estrecha vínculos físicos para fortalecerse, como así también emocionales y culturales, para afrontar las adversidades devenidas del deporte mismo. En el *framework*, Scrum toma esa metáfora y la lleva al campo organizacional para poder generar equipos interdisciplinarios que apunten a apoyarse mutuamente mediante la autoorganización y autogestión, de cara a maximizar la entrega de valor dentro de un proyecto. Más allá de su campo de acción, Scrum es un marco de trabajo y a la vez un modelo organizativo, que trasciende las actividades dentro de una compañía. Esta característica implica que puede aplicarse independientemente del sector del que surja la necesidad, o de la naturaleza del producto o servicio a desarrollar. También es importante aclarar que por tener esta cualidad, habilita el trabajo desde la agilidad en tres dimensiones: estructuras líquidas, procesos eficientes y cultura colaborativa.

Jeff Sutherland, uno de los creadores del marco Scrum explica en su libro que junto a Ken Schwaber lo concibió

Como una manera más rápida, confiable y eficaz de crear software en la industria de la tecnología (...) Hasta ese momento —y aun en 2005—, la mayoría de los proyectos de desarrollo de software se ejecutaban usando

el método en cascada, donde un proyecto se completaba en etapas separadas y avanzaba paso a paso hacia el lanzamiento último a los consumidores o usuarios de *software*. El proceso era lento e impredecible, y a menudo no daba como resultado un producto que la gente necesitara o pagara. Retrasos de meses o hasta años eran endémicos al proceso. Los primeros planes paso a paso, expuestos en cómodo detalle en gráficas de Gantt, daban seguridad a la dirección de que teníamos el control del proceso de desarrollo, pero, casi sin falta, pronto nos atrasábamos respecto al programa y rebasábamos desastrosamente el presupuesto. (Sutherland, 2016, p. 3)

Claramente, comenzamos a vislumbrar el porqué de Scrum y su creación. Partiendo de la necesidad de generar una respuesta adaptativa a un problema complejo desde varios frentes, la idea es flexibilizar el camino del proceso productivo para generar mejores contextos a medida que se avanza. Se entiende que debemos aceptar la incertidumbre.

Ya lo decía en la antigüedad el griego Heráclito: la única constante es el cambio. Esta frase puede servirnos de disparador para comenzar a pensarnos como seres ágiles. Analicémosla: el cambio, como concepto definido desde la RAE, es la “acción o efecto de cambiar” (s. f., definición 1). Usualmente, entendemos al cambio como un suceso disruptivo, que genera un vuelco en la mirada de una situación o elemento, dejando atrás un campo distinto, diferenciándose de lo nuevo que surge. Cambiar, entonces, se convierte en la acción o materialización de esa disrupción en un tiempo dado. En agilidad, esta idea toma un sentido diferente.

Decimos que el cambio es la única constante porque es casi con exclusividad, la única variable que podemos anticipar. Ser *Agile*, en este sentido, implica aceptar esa constante incertidumbre, aceptar que algo puede y que necesita transformarse para poder mantenernos dinámicos, permeables y, en resumidas cuentas, flexibles.

De esta reflexión, es que parten los conceptos más basamentales de la agilidad, lo que en el campo de acción acostumbramos a llamar **pilares** o *valores ágiles*. Son nada más y nada menos que la manera en la que esquematizamos el accionar ágil contribuyendo a la formación de su sentido sistémico y metodológico, como así también su contexto de aplicación definido. Los mismos aparecen dentro del llamado **Manifiesto ágil** en 2001, como iniciativa de homologación de la práctica

puntualmente dentro del desarrollo de *software*. 17 críticos de los modelos contemporáneos se reunieron puntualmente para resumir los principios que hoy conforman el mentado Manifiesto.

Figura 1: Manifiesto por el desarrollo ágil de *software*



Fuente: captura de pantalla de Agile Manifiesto (<https://agilemanifesto.org/iso/es/manifiesto.html>)

Se trata de cuatro valores y doce principios que organizan de manera un tanto holística lo que se espera que suceda dentro de un proceso de cambio institucional en todos sus niveles, con todos los participantes de esa organización dentro o “subidos al cambio”.

Los cuatro valores ágiles

1. “Individuos e interacciones sobre procesos y herramientas
2. Software/Producto funcionando sobre documentación extensiva
3. Colaboración con el cliente sobre la negociación contractual
4. Responder ante el cambio sobre seguir un plan” (Agile Manifiesto, s. f., <https://bit.ly/32RENOW>)

De lo expuesto anteriormente, se desprenden los doce principios ágiles que aportan más luz al camino de un proceso ágil.

- Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de productos que aporten valor.

- Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventajas competitivas al cliente.
- Entregamos un producto funcionando frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
- Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- El producto funcionando es la medida principal de progreso.
- Los procesos Ágiles promueven el desarrollo sostenible. Los interesados, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.

- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia. (Agile Manifiesto, s. f., <https://bit.ly/32REN0W>)

La guía Scrum: las reglas del juego

Como ya hemos lo hemos nombrado en párrafos anteriores, la agilidad se vio en la necesidad de homologar principios de acción frente al desarrollo de aplicativos. En este sentido, ponerse de acuerdo fue fundamental para generar la documentación que hoy se ha convertido en lecturas obligatorias para aquellos que quieran emprender el camino del agilismo. Entender que el campo de la agilidad demanda reflexiones casi constantes para alimentar la capacidad creativa, es la base para encarar tanto el ya nombrado *Manifiesto Agile* con sus valores ágiles y sus doce principios, como así también la guía Scrum.

Centrados puntualmente en esta última, sabemos que se trata de una compilación de preceptos que le dan propósito y plantean el andamiaje para todos aquellos que quisieran utilizar Scrum en su devenir laboral. Desde el punto de vista de sus orígenes, comenzamos en los primeros años de los 90 cuando Ken Schwaber y Jeff Sutherland desarrollan el marco aunque recién en 2010 lanzan la primera versión de *La guía definitiva de Scrum: las reglas de juego*.

Desde aquel momento hasta hoy, el conocimiento enmarcado dentro ha ido evolucionando, actualizando los saberes hasta la fecha. En ella, se da la definición de Scrum que ya hemos abordado, y también se explican los elementos que forman parte del *framework*, dándole sustento y sentido a cada uno de ellos, con el foco puesto en la entrega de valor a nivel global y los resultados.

El *framework*, ampliando su ámbito de aparición, surge como respuesta a una situación planteada dentro de los procesos de desarrollo IT que se apoyaban tradicionalmente en Waterfall. Como modelo productivo este se alinea con el ya nombrado control de procesos predictivos, generando ambientes controlados y con análisis exhaustivos para reducir riesgos y costos de producción. En este sentido, no se tiene en cuenta la necesidad del usuario final durante el proceso productivo, y solo hay *feedback* al final del mismo cuando el entregable se ha terminado, a diferencia de las metodologías ágiles (dentro de las cuales está Scrum) donde el plan es flexible y permeable a los cambios que puedan surgir, incluida la posibilidad de que los deseos del cliente cambien.

Sutherland en su libro *SCRUM: El arte de hacer el doble de trabajo en la mitad de tiempo* explica en un par de líneas por qué decide crear Scrum:

En 1993 inventé una nueva forma de hacer las cosas: Scrum, que representa un cambio radical respecto a las metodologías prescriptivas y verticales de gestión de proyectos del pasado. Scrum se asemeja, en cambio, a los sistemas evolutivos, adaptativos y capaces de autocorregirse. Desde su concepción, su marco se ha convertido en el modo en que la industria de la tecnología crea software y productos nuevos. (2016, p. 3)

¿Qué significa entonces esto? ¿Waterfall ha fallado y no sirve, y un marco de trabajo ágil como Scrum es la respuesta definitiva? De ningún modo. Ya dijimos antes, que el análisis de las variables que forman el contexto de acción es lo que dará la luz para saber qué marco de trabajo deberemos adoptar para lograr nuestros objetivos. Si tenemos horizontes con variables que pueden predecirse, y generar control de sus fluctuaciones cómodamente desde el punto de vista productivo, entonces nos encontramos frente a un proyecto que puede ser llevado adelante de manera tradicional. Sí, en cambio, tenemos ambientes complejos, con poca certeza y mucha incertidumbre, donde la práctica se deba ir descubriendo (emergente) a medida que avanzamos, entonces aquí estamos frente al campo de acción de Scrum, por ejemplo.

En búsqueda de una definición de Scrum

Adentrándonos ya en una definición propiamente dicha, según la Guía Scrum, se trata de “un marco de trabajo liviano mediante el cual las personas, los equipos y las organizaciones generan valor a través de soluciones adaptativas a problemas complejos” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>).

¿Qué significa que sea **liviana**? ¿Qué podemos inferir de esa frase, teniendo en cuenta lo expuesto hasta el momento? ¿Será que el *framework* SCRUM está incompleto? ¿Cómo es esto posible, y cómo puede ser que igualmente funcione?

Pensemos en la forma en la que Scrum se mueve. Proporciona una estructura mínima, incompleta, adrede, para poder construir de manera evolutiva dentro de ella. Es por ello que debe ser complementado con técnicas y prácticas específicas de la industria o campo donde se esté llevando a cabo el desarrollo del producto, dado que el “cómo se hace” del proceso viene directamente de la experiencia y devenir del equipo. Por eso es **liviana**.

El enfoque está dado por la entrega de valor rápida, frecuente. Quienes trabajan con Scrum se aseguran de que el cliente vea funcionando al producto, a medida que se desarrolla y evoluciona (recordemos la frase “el cliente dentro”). Así es como

el equipo puede inspeccionar también la experiencia del usuario y la suya misma, para adaptar lo que sea necesario, aprendiendo en el camino. Recapitulemos en algo que no debemos olvidar: entre las cuestiones más importantes a tener en cuenta a la hora de implementar Scrum, deberemos estar seguros de que el análisis de los datos recabados para desarrollar el producto esté enmarcado dentro de lo que venimos charlando como entornos complejos, inciertos, donde además de la necesidad de entrega de un producto cualquiera, también es menester ir descubriendo en paralelo lo que construiremos.

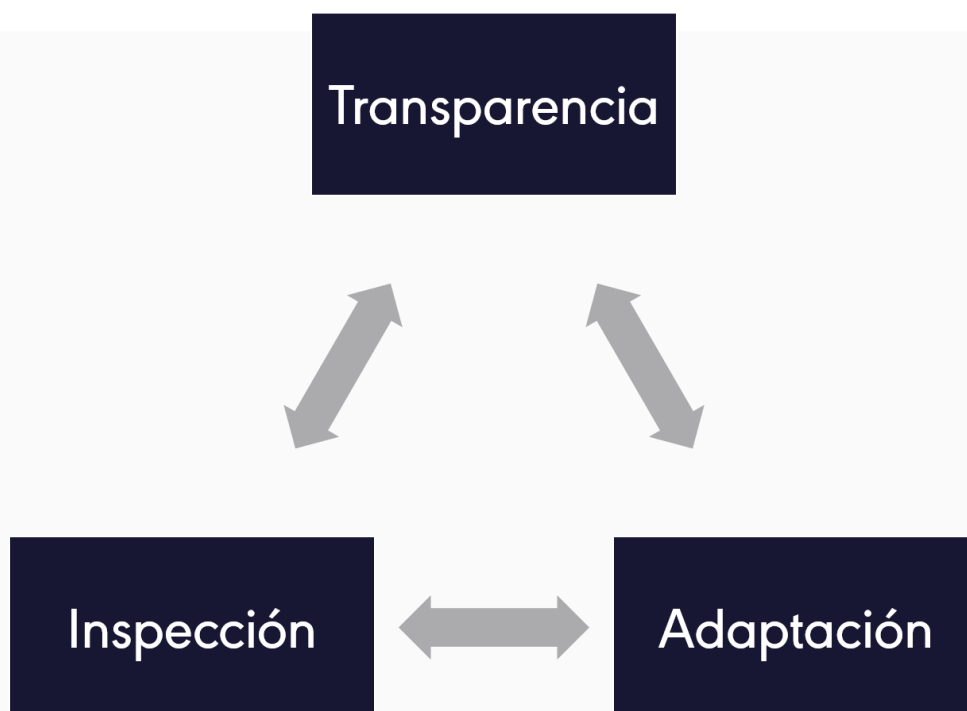
Las condiciones cuando se utiliza Scrum posibilitan la creación de soluciones innovadoras, que se pueden adaptar en función de lo que se aprende mientras se desarrolla. Aun así, no debemos confundir esto con pensar que el marco por sí solo gestiona proyectos o equipos. Puede ser confuso en un principio, pero lo importante aquí es entender que Scrum se basa en la **autogestión** del equipo y la **autoorganización**: el equipo tiene completa autonomía para ordenar cómo, quién y en qué estará trabajando cada uno.

No tiene estructuras minuciosamente definidas (procesos, subprocesos, etc.) o detalladas, por lo dicho anteriormente respecto a estar incompleto a propósito y de no buscar predecir propiamente dicho; no se busca que las tareas estén detalladas día a día, porque el marco está enfocado en solventar la incertidumbre, y la incertidumbre plantea la posibilidad de que en cada día surjan tareas nuevas y desafiantes.

Centrémonos ahora en la forma en la que Scrum se mueve, y hace que el equipo evolucione. Retomando lo expresado respecto del empirismo, el marco está apoyado en tres pilares: trabaja fuertemente en **transparentar, inspeccionar y adaptar** todos los procesos y decisiones que surjan de la misma práctica.

Es esto justamente lo que en principio lo vuelve netamente empírico. Porque **transparenta** las acciones dentro de ciclos cortos de trabajo (*sprints*), para posibilitar la **inspección** de los resultados. Esto último da la posibilidad de generar **adaptación** si es necesario para responder siempre a las necesidades y condiciones cambiantes que vayan surgiendo.

Figura 2: Transparencia, inspección y adaptación



Fuente: elaboración propia.

En *La guía Scrum* se indica que “Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir tales habilidades según sea necesario” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>). La forma en la que logren, va a depender de las interrelaciones que se generen dentro del equipo, y cómo utilizan los espacios donde transparenten, generen inspección y adapten dentro de los eventos que componen el circuito Scrum.

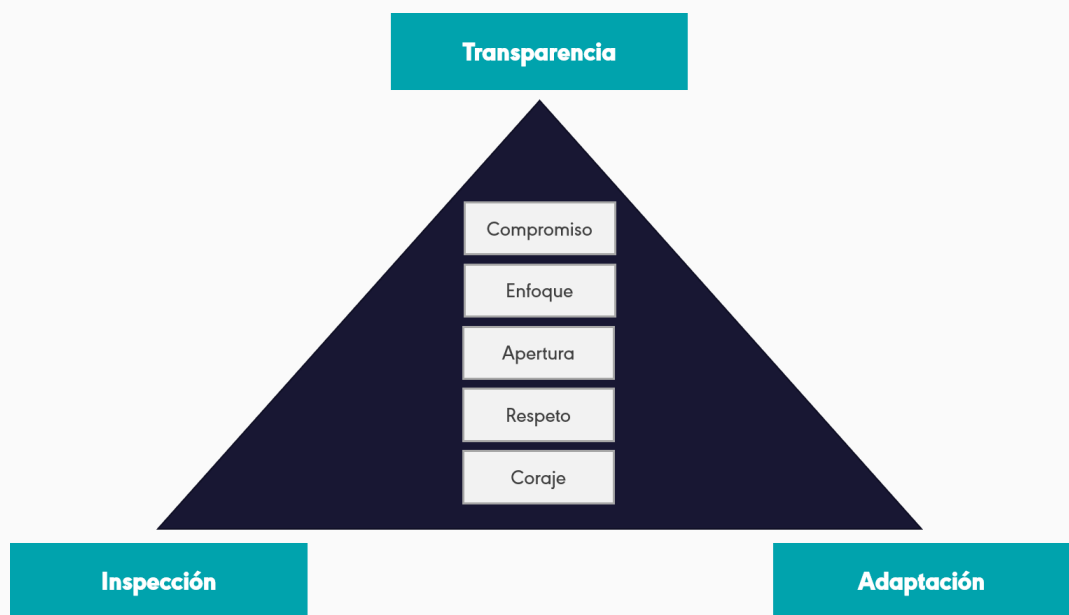
Las personas son el principal factor de éxito a la hora de la creación de productos. Es más importante construir un buen equipo que construir el contexto. Muchas veces se comete el error de construir primero el entorno de trabajo y esperar a que el equipo se adapte. Por entorno me refiero a la oficina, los escritorios, los computadores, el *software*, los procesos de trabajo, etc. Por el contrario, la agilidad propone crear primero el equipo y que este construya su propio entorno y procesos con base en sus necesidades.

Para poder llevar adelante exitosamente lo requerido, es necesario que cada integrante del equipo pueda cultivar como mínimo cinco puntales importantísimos: **compromiso, enfoque, apertura, respeto y coraje**, los cuales hasta podrían ser tomados como fundacionales para la tarea de **transparentar-inspeccionar-adaptar**

que plantea Scrum como pilares. Sin estos valores o verticales, el equipo podría flaquear.

De utilizar Scrum, el equipo se debe preocupar por cultivar estos pilares, con el objetivo de mejorar su capacidad autogestiva. De su sinergia, el crecimiento colaborativo y la utilización de los aprendizajes surgidos de la inspección continua en sus eventos, depende el éxito y la consecución de los objetivos planteados.

Figura 3: Éxito y consecución de los objetivos planteados



Fuente: elaboración propia.

El equipo de Scrum se **compromete** a lograr sus objetivos y apoyarse mutuamente. Su **enfoque** principal es el trabajo del Sprint para hacer el mejor progreso posible hacia estos objetivos. El equipo de Scrum y sus partes interesadas están **abiertos** sobre el trabajo y los desafíos. Los miembros del equipo de Scrum se **respetan** mutuamente para ser personas capaces e independientes, y son respetados como tales por las personas con las que

trabajan. Los miembros del equipo de Scrum tienen el **valor** de hacer lo correcto y de trabajar en problemas complejos.

Estos valores dan dirección al equipo de Scrum con respecto a su trabajo, acciones y comportamiento. Las decisiones que se toman, las medidas tomadas y la forma en que se utiliza Scrum deben reforzar estos valores, no disminuirlos o socavarlos. [...] .. Cuando estos valores son asimilados por el equipo de Scrum y las personas con las que trabajan, los pilares empíricos de Scrum de transparencia, inspección y adaptación cobran vida construyendo confianza. (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

El corazón de Scrum: eventos

Teniendo en cuenta lo anterior, los equipos trabajan dentro de un evento contenedor llamado **sprint**, donde suceden la mayor parte de los sucesos. El *sprint* es donde el equipo se mueve y opera bajo los pilares y valores nombrados. Se lo considera el latido del corazón de Scrum, y a su vez es un evento contenedor, porque dentro de él acontecen otros eventos, de suma importancia para el desarrollo cabal del *framework*.

Puntualmente, Scrum delimita cinco ceremonias o eventos que operan para que se dé el control de sus procesos. Para que Scrum funcione, deben aplicarse fielmente, de lo contrario podemos estar debilitando la posibilidad de éxito del marco. Mediante estas ceremonias, Scrum se asegura de mantener los mínimos necesarios para facilitar que el control empírico de procesos funcione (transparentar - inspeccionar - adaptar).

Como indica la misma *Guía Scrum*

Cada elemento del marco sirve a un propósito específico que es esencial para el valor global y los resultados realizados con Scrum. Cambiar el diseño o las ideas básicas de Scrum, dejar fuera los elementos, o no seguir las reglas de Scrum, cubre los problemas y limita los beneficios de Scrum,

potencialmente incluso haciéndolo inútil. (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

Ya nombramos que Scrum plantea que el proceso productivo sea **iterativo e incremental**, trabajando en ciclos cortos (cuando uno termina, inmediatamente comienza otro) que puedan empíricamente analizarse con sus pilares. Esos ciclos cortos de trabajo son ni más ni menos que los *sprints*. Estos se mueven en forma de bucle, e involucran dentro de su devenir, otros eventos antes de su comienzo, durante y posteriormente a que este termina. De la última frase, es importante entender que el *sprint* en Scrum es un evento continuo, es decir, que su duración no cambia mientras se desarrolla el producto, permitiendo de esta forma que se cumpla el circuito empírico de los valores de Scrum.

La duración de un Sprint, como máximo puede ser de cuatro semanas. Puede ser menos, pero no más. Todo depende de la finalidad que el equipo persiga, dado que es el mismo equipo el que debe descubrir cuál es el periodo de tiempo mínimo que requiere para generar valor y realizar la entrega de un incremento del producto terminado. Aquí se introduce el concepto de *time box* dentro del *sprint*, que implica mantener tiempos específicos y constantes para poder definir el ritmo de la iteración dentro de Scrum. El *time box* es una constante dentro de Scrum, y la *Guía Scrum* específica en cada apartado cuánto debería durar cada evento dentro del marco teniendo como variable su duración dentro del ciclo de trabajo.

Como evento, el *sprint* posibilita dentro de su devenir, el suceso de otras ceremonias que ayudan al equipo a traccionar el objetivo. Eventos donde se practican de alguna manera los tres pilares, transparencia-inspección-adaptación, y se relacionan con alguna de las dos vías: evolución del producto o mejora continua del proceso.

- *Sprint planning*

Este evento es anterior al comienzo de un *sprint* cuando el proyecto está comenzando, es decir, puede convertirse en la planificación anterior al *kick off* o arranque. Ya dentro del circuito iniciado, se convierte en una ceremonia que ocurre en cada principio de *sprint*. Cuando un *sprint* termina, inmediatamente otro comienza con la planificación de los trabajos a realizar. Dentro de ella, todo el equipo Scrum se pone de acuerdo mediante el trabajo colaborativo y crea un plan a llevar adelante durante el siguiente *sprint*.

Si bien es un evento de equipo, es posible que se inviten asesores externos para facilitar el planeamiento. Lo que se intentará es buscar la forma en la que la planificación aporte el mayor valor y utilidad de cara al próximo ciclo corto de trabajo y así alcanzar el objetivo de ese *sprint* (delimitado

por el equipo Scrum). Cuanta más información se tenga respecto del rendimiento y capacidad del equipo, mejor se podrá planificar el esfuerzo.

- *Daily Scrums*

Son ceremonias de sincronización de los equipos Scrum. En ella mínimamente los participantes contestan tres preguntas para informar qué hicieron ayer, qué impedimentos tienen y qué harán hoy. De esta forma, el equipo no solo conoce y se interioriza de lo que está haciendo cada uno, sino que también puede solventar internamente las trabas que se presenten. A su vez, también se inspecciona el progreso hacia el objetivo del *sprint* y, ante cualquier eventualidad, puede ajustarse el trabajo a realizar. El objetivo del *sprint* es crear un incremento de producto utilizable y potencialmente desplegable para que nuestro cliente pueda dar *feedback* y utilizar nuestro producto.

Con una duración de 15 minutos como máximo, de hecho suele llamarse *daily stand up*, para que sea una reunión tan breve que pueda llevarse a cabo de pie. Se busca ser puntual e involucrar a los desarrolladores del equipo Scrum, siempre en el mismo lugar durante todos los días de trabajo planteados.

- *Sprint review*

Es un evento de inspección del incremento realizado (el entregable funcional). Aquí

El propósito de la revisión del *sprint* es inspeccionar el resultado del *Sprint* y determinar futuras adaptaciones. El equipo de Scrum presenta los resultados de su trabajo a las partes interesadas clave y se discute el progreso hacia el objetivo de producto. (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

La importancia está dada por ser uno de los ámbitos de *feedback* y contacto con nuestros clientes o usuarios, para saber si lo que estamos realizando es realmente valioso para ellos, o debemos adaptar algo en particular para responder mejor a sus necesidades.

No es una demostración del producto funcionando. Tampoco es un ámbito de felicitaciones por el desarrollo generado. Es en resumidas cuentas es una reunión de trabajo y, como tal, es importante entender que cobra sentido para enmarcar la estrategia del negocio puntualmente.

- *Sprint retrospective*

Este evento aparece como el espacio de reflexión propiamente dicho, con el propósito de buscar formas de aumentar la calidad y la eficacia del grupo. La idea es que el equipo Scrum inspeccione el último *sprint* para poder buscar esas respuestas. Es importante tener en cuenta que en este punto dependiendo del dominio en el cual estemos trabajando, puede cambiar el objeto de análisis del espacio. Aquí es donde se explora los orígenes de las desviaciones, se identifican y se plantean accionables para intentar no repetirlos en el ciclo siguiente. También se apuntala lo realizado positivamente para replicarlo, los problemas encontrados para saber cómo se resolvieron si así fue. Es el punto final de un *sprint*.

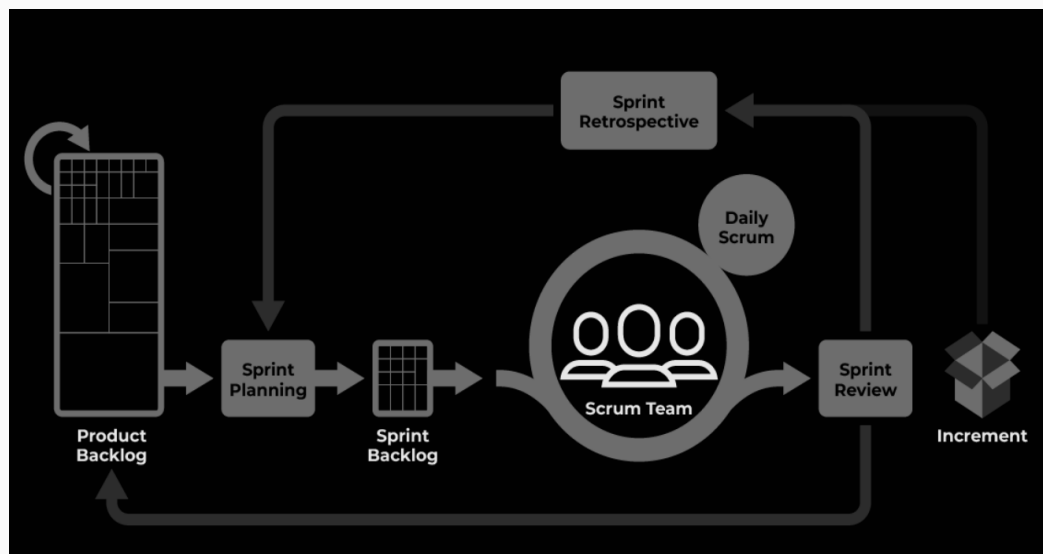
- *Grooming or refinement*

Como adicional, podemos contar con este evento, que ocurre dentro del *sprint*, pero que es más bien una **práctica recomendada**. El *grooming* o en español refinamiento es una práctica que ayuda a estar preparado. Todo el equipo Scrum participa, e incluso puede sumarse alguien externo que aporte soluciones a los temas que quieran tratar.

Vital importancia para que este espacio tenga el impacto necesario: todos los presentes deben conocer los requerimientos o temas a tratar y se espera que todos los que asistan y aporten su presencia sean relevantes respecto del objetivo del espacio.

De manera simple y acotada, esto es lo que compone un ciclo en Scrum. Todo lo que sucede, sucede dentro del *sprint*, y cada *sprint* debe tomarse como un proyecto en sí mismo. Esto es algo complejo de entender y, sobre todo, de internalizar en aquellos equipos que están comenzando a trabajar con el marco recientemente. Es interesante que podamos ver al *sprint* como un “proyecto” que empieza y termina iterativamente, por lo que todas las tareas necesarias que se comprometieron dentro del mismo siempre estarán orientadas a generar el máximo de valor. En este sentido, el mismo Sutherland, uno de los creadores de Scrum, amplía el horizonte de interpretación sumando “los conceptos de mejora continua y productos mínimos viables para obtener realimentación inmediata de los consumidores, en lugar de esperar hasta que un proyecto concluya” (2016, p. 3).

Figura 4: Ciclo Scrum. Circuito completo con todas sus ceremonias y artefactos



Fuente: elaboración propia.

El equipo Scrum

Según la *Guía Scrum*, el equipo es una unidad fundamental. La suma de sus roles da como resultado el **equipo Scrum**, por lo que se entiende que todos forman parte de la misma unidad: los desarrolladores (como sinónimo de todos aquellos individuos que trabajan dentro del proyecto analizando, instrumentando, desarrollando, implementando, probando, etc.), el *Scrum Master* (como guía y guardián de la práctica y el marco) y el *Product Owner* (como encargado de recopilar y ordenar las prioridades que requiere el desarrollo del producto).

Un factor importante para considerar relacionado con el tamaño es que se espera que dentro de un equipo Scrum se cuente con todas las habilidades necesarias para poder entregar incrementos de valor al final de cada *sprint*. Esto implica que no necesitar la intervención de personas externas al equipo. Esta característica la llamamos multifuncionalidad. El equipo Scrum es un equipo **multifuncional**. Todos los involucrados trabajan y empujan hacia un objetivo común, que es la creación de incrementos (valor) del producto a desarrollar.

Nadie externo al equipo interviene decidiendo tampoco qué trabajo realizan, cuándo lo realizan ni cómo lo realizan. Las decisiones sobre estos tres aspectos son potestad exclusiva del equipo Scrum. El equipo Scrum decide el qué, el cuándo y el cómo. Esta característica la llamamos **autogestión**. El equipo Scrum es un equipo autogestionado.

Apelando a su capacidad multifuncional y de autogestión, el equipo Scrum es responsable de hacer todo lo que sea necesario para entregar un incremento al

final de cada *sprint*. Esto no solo incluye la construcción del incremento, sino que también abarca la colaboración con tus *stakeholders*, la investigación, la experimentación, la negociación, la capacitación, el aprendizaje, etc.

Una vez más, nos encontramos haciendo la salvedad de que Scrum, como marco de trabajo, no tiene como objetivo gestionar al equipo. Scrum sucede en el equipo, y esto solo puede ser así si el equipo mismo se compromete desde su rol a llevar adelante el marco como tal:

Cada elemento del marco sirve a un propósito específico que es esencial para el valor global y los resultados realizados con Scrum. Cambiar el diseño o las ideas básicas de Scrum, dejar fuera los elementos, o no seguir las reglas de Scrum, cubre los problemas y limita los beneficios de Scrum, potencialmente incluso haciéndolo inútil. (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

Figura 5: Equipo Scrum



Fuente: elaboración propia.

Como reflexión de esta parte es importante entender que dentro de Scrum no se reconocen las jerarquías. Típicamente, este equipo estará constituido por una cantidad no mayor o igual a diez personas. Es lo ideal. Una cantidad más grande puede no ser lo suficientemente eficiente si se debe cambiar de dirección para tomar decisiones de manera ágil. Por otro lado, una cantidad menor, puede no bastar para generar un incremento de valor lo suficientemente considerable dentro del *sprint*, teniendo en cuenta no solo el esfuerzo, sino también la multifuncionalidad necesaria para solventar los bloqueos dentro del equipo, aumentando las dependencias del exterior.

Existe la posibilidad de que si el equipo es muy grande, pueda dividirse en equipos más pequeños de aproximadamente diez personas, todas trabajando en un mismo producto y todas compartiendo no solo el mismo *Product Owner* sino con un mismo objetivo de producto y *Product Backlog*. En agilidad, esto se llama LeSS o Scrum a gran escala.

Cabe destacar que nadie del exterior debe intervenir en las decisiones del equipo Scrum, ni cómo realizan las tareas. Todo esto es del equipo, y solo del equipo, apelando a su capacidad de autoorganización y autogestión.

- **Los desarrolladores**

Son todos aquellos individuos (analistas, científicos, especialistas, etc.) que hacen el trabajo dentro del marco. La Guía usa el término desarrolladores no para excluir, sino para simplificar. Se comprometen a crear **incrementos útiles** (funcionales) en cada *sprint*.

Se encargan de desarrollar el producto, se autoorganizan y autogestionan, para lograr un entregable que cumpla con los criterios de calidad al final de un ciclo de trabajo (*sprint*) con los elementos tomados del *Product Backlog* o lista de trabajo pendiente del producto. En la mayoría de los casos se trata de equipos interdisciplinarios, capaces de organizarse para no tener dependencias externas.

- **El Scrum Master**

Es el guardián del marco Scrum, tal y como su traducción lo indica. Es responsable de honrar el conocimiento que establece la *Guía Scrum*, y es el encargado de que el mensaje vertido sea claro y llegue sin problemas tanto al equipo como a toda la organización asegurando la efectividad de las prácticas dentro del marco y liderando servilmente.

Podríamos decir entonces que el *Scrum Master* tiene dos funciones puntuales. Por un lado, gestionar el marco y, por el otro, eliminar impedimentos de manera progresiva y constantemente conforme vayan surgiendo tanto en el equipo como en la organización. En resumidas cuentas, el *Scrum Master* debe ser el responsable de velar porque Scrum se lleve adelante, transmitiendo sus beneficios a la organización facilitando su implementación.

Con respecto a la *Guía Scrum*, los autores resumen el actuar del *Scrum Master* en dos grandes momentos.

Colaboración con el *Product Owner*:

- Ayudar a encontrar técnicas para una definición eficaz de los objetivos del producto y la gestión de los retrasos en el producto.
- Ayudar al equipo de Scrum a comprender la necesidad de elementos de trabajo pendiente de productos claros y concisos.
- Ayudar a establecer la planificación empírica de productos para un entorno complejo.

- Facilitar la colaboración de las partes interesadas según sea solicitado o necesario.

Colaboración con la organización:

- Liderar, capacitar y mentorizar a la organización en su adopción de Scrum.
- Planificar y asesorar sobre la implementación de Scrum dentro de la organización.
- Ayudar a las personas y a las partes interesadas a comprender y promulgar un enfoque empírico para el trabajo complejo.
- Eliminar las barreras entre las partes interesadas y los equipos de Scrum.

- **El Product Owner**

El *Product Owner* (PO) es el único responsable de tomar decisiones acerca del producto. El producto puede ser un producto digital, físico, un servicio o, inclusive, algo más abstracto como una experiencia. Aunque puede recibir ayuda de otros, el *Product Ownership* no es un trabajo que realiza un comité, es fundamental que esté encarnado en una sola persona. Las responsabilidades de *Product Owner* son incompatibles con las del *Scrum Master* o con las de un desarrollador, por más que en algunos equipos vemos que realiza diversas tareas, por su conocimiento o por falta de recursos.

El PO gestiona el flujo de valor de producto desde el *Product Backlog* y, además, es el interlocutor con los *stakeholders* y *sponsors* del proyecto, como así también de los clientes. Por esta razón es importante que detente el poder necesario para que esto suceda. Se puede decir que es un gestor de requisitos o un cliente que se encarga de gestionar el *Product Backlog* aunque esta definición estaría quedándose corta. Porque no solo tiene la responsabilidad de que el *Product Backlog* esté bien armado, detallado y priorizado, sino que también debe entender perfectamente cuál es el horizonte que se plantea para el desarrollo del producto.

El trabajo del *Product Owner* se desarrolla durante todo el día, todos los días del *sprint*. Es parte del equipo Scrum, está en contacto constante con el resto del equipo, aun si trabajan de forma remota, está accesible todo el tiempo. Si trabajan de manera presencial, se sientan junto a los desarrolladores. En cualquier caso, trabaja codo a codo con el resto del equipo Scrum durante todo el *sprint*.

Ya hemos hablado de la importancia de la transparencia, como uno de los pilares de Scrum. En esta ocasión explicaremos cómo contribuye el PO a esta transparencia: visibilizando el *product backlog*. El PO es responsable de que

el *Product Backlog* sea accesible y conocido por todos los involucrados, ya sean miembros del equipo Scrum, *stakeholders* o cualquier persona en la organización. Esto no significa que cualquiera pueda alterarlo, pero sí debes facilitar el acceso a verlo, con el fin de informarse. Por ejemplo, sin importar la herramienta que utilicen para almacenar el *Product Backlog*, la cual podría ser Excel, Google Spreadsheets, Trello, Monday, Jira, etc.; no haya restricción de acceso y en cada comunicación/*email* que se envíe podría incluir en el pie un *link* a la visualización del *Product Backlog*.

Por último, en cada *sprint*, el *Product Owner* marca el *sprint goal* de acuerdo con el equipo de desarrollo, teniendo en cuenta lo que aporte más valor al producto y que genere incrementos constantes.

- Los *stakeholders*

Es importante entender que los *stakeholders* son importantes en Scrum por tener influencia sobre el accionar del marco, pero no por participar activamente en él como sí lo hacen los roles anteriormente nombrados. Son llamados así o también grupos de interés, entre otras acepciones. Se hace referencia a todas aquellas personas, organizaciones o empresas que apoyan el desarrollo de un producto. Hacen que una organización tenga sentido de existencia aun cuando no participen activamente en ella o tengan relación directa con el desarrollo de los servicios o productos que la organización desarrolla (clientes finales, accionistas, proveedores, otros trabajadores de la misma empresa en diferentes departamentos, medios de comunicación, etc.).

Es muy importante tener bien sabido quiénes son nuestros *stakeholders*, dado que son los que se verán afectados por nuestras decisiones y actividades respecto del producto. Mediante su *feedback* enmarcado dentro de los eventos de Scrum (*sprint review*) y la comunicación con el *Product Owner*, el equipo Scrum puede recolectar valiosos datos para aprender de las necesidades y deseos de los *stakeholders*, y así canalizarlos a través del *Product Backlog*.

Artefactos de Scrum

“Los artefactos de Scrum representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave. Por lo tanto, todos quienes los inspeccionan tienen la misma base para la adaptación” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>).

En Scrum, se denomina **artefactos** a aquellos elementos físicos que se generan dentro de la aplicación del marco mismo y contienen información. Cada artefacto reúne y

visibiliza toda la información necesaria para tomar decisiones periódicas de adaptación correspondientes a un compromiso en particular.

Tenemos tres artefactos que, según la *Guía Scrum*,

Representan trabajo o valor. Están diseñados para maximizar la transparencia de la información clave. Por lo tanto, cada uno de los que los inspecciona tienen la misma base para la adaptación. Cada artefacto contiene un compromiso para garantizar que proporciona información que mejora la transparencia y el enfoque con el que se puede medir el progreso.

(Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

Los enumeramos a continuación:

- **Product backlog**

Es básicamente un inventario que contiene los trabajos a realizar para llevar adelante el producto: tareas, dependencias, casos, requerimientos, etc. Según la *Guía Scrum*, “es una lista emergente y ordenada de lo que se necesita para mejorar el producto” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>).

Se trata de la más importante fuente de información sobre el producto en Scrum, independientemente del formato. Esta lista es el resultado de la gestión exclusiva que el *Product Owner*, como encargado, va formando en conjunto con el cliente, los *stakeholders*, etc. Refleja lo que falta por hacer para implementar el producto, como también lo ya realizado. Como tarea adicional, el *Product Owner* debe priorizar lo importante y qué entrega más valor en la lista en cada etapa, detallar puntualmente para que los desarrolladores sean capaces de interpretarlo y ejecutarlo. En la Guía se lo nombra como refinamiento: el acto de descomponer y definir aún más los elementos de trabajo pendiente del producto en artículos más pequeños y precisos. Esta es una actividad en curso para agregar detalles, como una descripción, un pedido y un tamaño. Los atributos a menudo varían con el dominio del trabajo. Puede contener funcionalidades, requisitos, historias de usuario (forma de expresar las tareas a realizar teniendo como base el criterio de quien usará el producto), tareas técnicas, resultados de investigación, entre otras.

Del *Product Backlog* se desprende el **objetivo del producto** como **compromiso**: describe un estado futuro del producto que puede servir como objetivo para el equipo Scrum contra el cual planificar. El objetivo del producto se encuentra en el trabajo pendiente del producto (*Product Backlog*). El resto del trabajo pendiente del producto surge para definir qué cumplirá el objetivo del producto.

- **Sprint Backlog**

Este artefacto se desprende puntualmente del *Product Backlog*. Se trata de una lista de elementos con los cuales se trabajará en el *sprint* actual. Normalmente, están compuestos por tareas más pequeñas que permiten conseguir un incremento del producto al finalizar el ciclo. Lo generan los desarrolladores y es una imagen muy visible y en tiempo real del trabajo que los desarrolladores planean realizar durante el Sprint para lograr el **objetivo de sprint**. Su razón de ser es poder mantener empíricamente la transparencia dentro del devenir del desarrollo. Y se actualiza durante toda la iteración, especialmente a través de las *Daily Scrums*.

Así como el *Product Backlog* posibilita la visibilidad de trabajo en progreso y lo que queda por hacer a nivel proyecto, el *Sprint Backlog* logra el mismo efecto, pero a nivel *sprint*, dando información detallada de quién está con cada tarea, qué esfuerzo le llevará, si tiene bloqueos, dependencias, etc. Cuanto más detallado esté, mejor será el nivel de comprensión del equipo al observar.

Aquí, el **compromiso** está dado por el **objetivo del sprint**, que es creado en el evento *Sprint Planning* como un compromiso de los desarrolladores para cumplir dentro del ciclo de trabajo. Si bien el *Sprint Backlog* puede sufrir modificaciones merced al devenir del *sprint* mismo y las negociaciones entre los desarrolladores y el *Product Owner*, el *Sprint Goal* u objetivo de *sprint*, no debe verse afectado.

- **Incremento**

Según la *Guía Scrum*, un incremento es “un paso de hormigón hacia el Objetivo de Producto” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>). En resumidas cuentas, un incremento es el resultado esperado de un *sprint*, puesto a disposición del usuario, aportando un valor de negocio al producto que se está desarrollando.

Es la suma de todas las tareas llevadas a cabo para generar un entregable dentro del *sprint*. Es importante entender, que en Scrum un incremento es considerado como tal, si toma en cuenta las funcionalidades de su predecesor como punto de partida siendo integradas satisfactoriamente. Es

por eso que se dice que los incrementos deben ser aditivos unos con otros y necesariamente debe poder usarse.

Dentro del *sprint*, puede haber varios incrementos, los cuales se estarán presentando dentro del evento *Sprint Review*, no siendo esto una condición excluyente para su entrega temprana, ya que este evento no es considerado el único momento donde se pueda entregar valor en Scrum. El trabajo no se considera parte de un incremento a menos que cumpla con la **definición de hecho**.

Esto último es el **compromiso** del incremento. Según la Guía, la

Definición de hecho es una descripción formal del estado del Incremento cuando cumple con las medidas de calidad requeridas para el producto. En el momento en que un elemento de trabajo pendiente de producto cumple con la definición de hecho, se crea un incremento. (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>)

Aquí se cumple el criterio de transparencia, puesto que si un trabajo pendiente no está cumpliendo con lo definido, no se puede lanzar ni presentar en la *Sprint Review*; vuelve al *Product Backlog* y retoma el curso natural del *sprint*.

Si bien estos son los tres artefactos principales del marco Scrum, y son considerados imprescindibles junto con sus compromisos, hay otros conceptos que entran en juego y que a pesar de no formar parte del núcleo, son importantes para asegurar la calidad y el éxito de Scrum.

- **Definición de listo** (*DoR - Definition or Ready*)

Se trata de un documento que delimita cuándo un requerimiento (una historia de usuario o similar) es considerado como **listo** para que el equipo de desarrollo pueda tomarlo, valorarlo, entenderlo e incluirlo en una *Sprint Planning* de cara a trabajarlo dentro del *sprint*.

- **Burndown chart**

Básicamente, se trata de una expresión gráfica del trabajo pendiente a lo largo del tiempo, y que ayuda a mostrar la velocidad que el equipo tiene para completar objetivos, requisitos o historias de usuario. Permite anticipar si el equipo podrá o no completar el trabajo en un tiempo estimado.

En los módulos siguientes exploraremos más a fondo la forma en la que estos conceptos toman sentido dentro de la práctica del marco, y cómo cada uno de ellos aporta valor al objetivo del producto.

Por último, y a modo de resumen, te propongo que analicemos la definición de Scrum identificando que sí es Scrum y que no es Scrum.

¿Qué sí es Scrum?

- “La unidad fundamental de Scrum es un pequeño equipo de personas, un Equipo Scrum. El equipo Scrum consta de un *Scrum Master*, un *Product Owner* y desarrolladores” (Schwaber y Sutherland, 2020, <https://bit.ly/3FPUAD4>).
- Scrum sí es un marco de trabajo liviano que proporciona una estructura mínima para poder construir una solución de forma evolutiva.
- Scrum es un diseño deliberadamente incompleto y minimalista. Cualquier persona, equipo u organización que desee crear productos de calidad con Scrum deberá complementarlo con técnicas y prácticas específicas de su industria. Scrum te dice qué hacer y no te dice cómo hacerlo.
- La generación de valor a través de Scrum implica la creación de un producto de forma tal que lo puedas ver funcionando muy frecuentemente a medida que crece y evoluciona. De esta manera, lo podrás inspeccionar y adaptar todo lo que consideres necesario.
- Este enfoque solo tiene sentido si estás frente a problemas complejos e inciertos, donde no solo debes construir un producto, sino también descubrir en paralelo qué producto construir.

¿Qué no es Scrum?

- ★ Scrum no es un enfoque para gestionar proyectos ni equipos.
- ★ En Scrum no hay gerentes que dirijan y controlen el trabajo que realizan las personas del equipo. Específicamente en este punto, Scrum se basa en la capacidad de autogestión del equipo, quien tiene completa autonomía para elegir en qué trabajar, cómo hacerlo y quiénes lo hacen dentro del equipo.
- ★ Scrum no es una metodología con procesos, subprocesos, *inputs* y *outputs*.
- ★ Scrum no es un método para hacer seguimiento detallado, día a día, de las tareas individuales de los miembros del equipo Scrum.
- ★ Scrum no sirve si lo que quieres es seguir haciendo lo mismo, pero más rápido.

Referencias

Agile Manifiesto (s. f.). *Manifiesto por el Desarrollo Ágil de Software*.
<https://agilemanifesto.org/iso/es/manifesto.html>

Schwaber, K.; y Sutherland, J. (2020). *La Guía Scrum*.
<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

Sutherland, J. (2016). *SCRUM: El arte de hacer el doble de trabajo en la mitad de tiempo*. Océano.