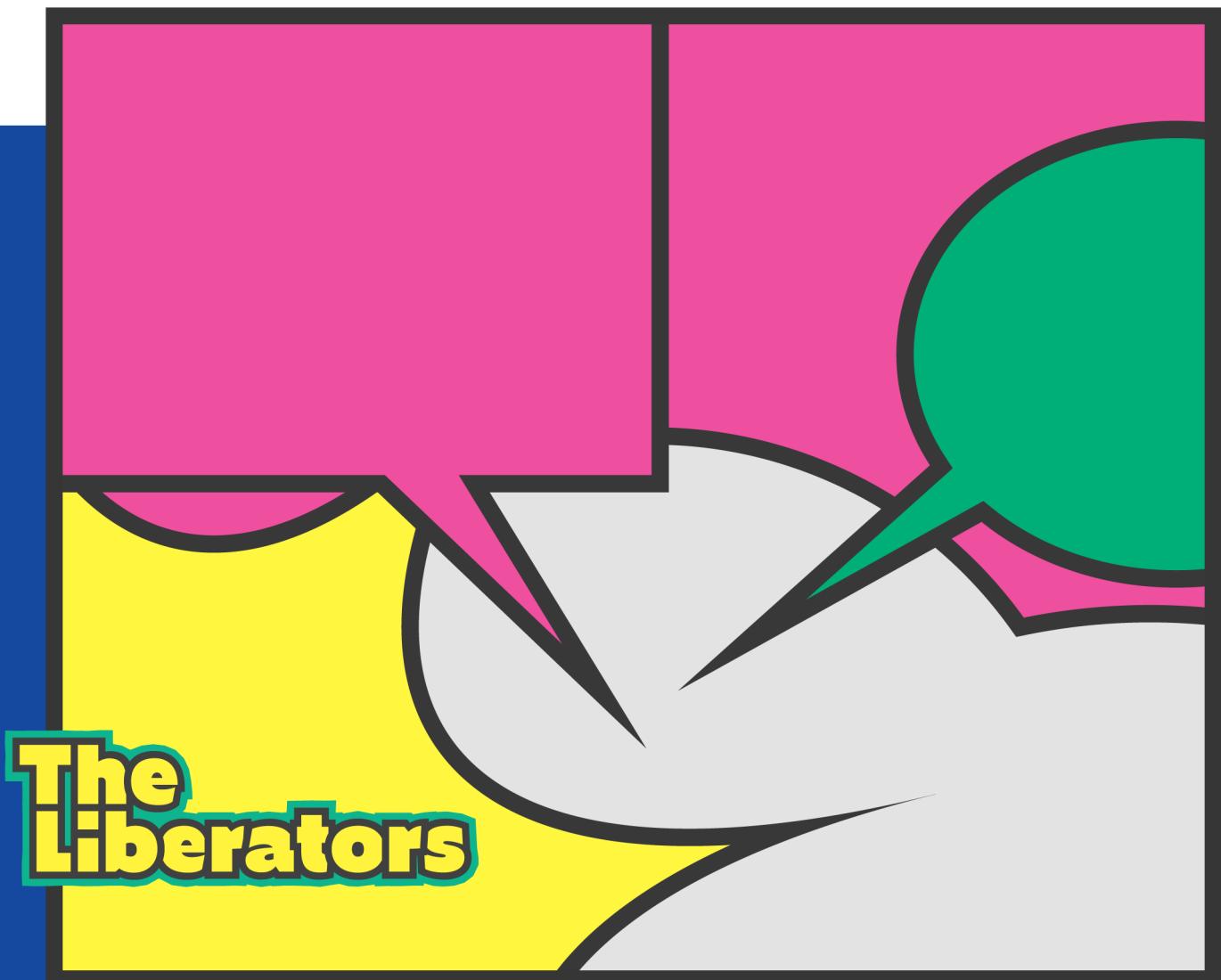


# El Marco De Trabajo Scrum: Un Marco De Trabajo Para Reducir El Riesgo Y Para Entregar Valor Antes

Una descripción general del marco de trabajo Scrum, para principiantes y para aquellos que deseen refrescar su conocimiento.



# En este Whitepaper



---

## **Prólogo**

---

### **Un Poco de Historia**

---

### **Los Pilares del Empirismo**

---

### **Scrum: Un Marco de Trabajo para el Control del Proceso Empírico**

---

### **Los Artefactos de Scrum**

---

### **Los Eventos de Scrum**

---

### **Las Responsabilidades de Scrum**

---

### **Dos Principios Rectores**

---

### **Cinco Valores Fundamentales que Permiten el Empirismo**

---

### **Todo Junto, un Marco de Trabajo Sencillo**

---

## Prólogo



«No nos importa *Scrum*», es lo que a veces le decimos a la gente. Eso, evidentemente, resulta sorprendente. Pero es nuestra forma de decir que lo que importa no es el marco de trabajo *Scrum*, sino lo que este hace posible. Cuando adoptas esa perspectiva, muchas preguntas teóricas se vuelven obvias o incluso dejan de tener sentido. Como, por ejemplo, «¿deberían estar todos los elementos del *Sprint Backlog* relacionados con el *Sprint Goal*?», «¿debería contener bugs el *Product Backlog*?» o «¿debo terminar la *Daily Scrum* exactamente a los 15 minutos?». Es fácil quedar atrapado en los pequeños detalles cuando se pierde de vista (o no se ve) el cuadro completo.

Nos encanta el marco de trabajo *Scrum* por lo que hace posible. La guía oficial de *Scrum* ya realiza un gran trabajo explicándolo de manera concisa. Nosotros queremos añadir una explicación con nuestras propias palabras, tal como lo hacemos en nuestro trabajo con Equipos *Scrum*, y como Entrenadores Profesionales de *Scrum* para Scrum.org. Nuestro objetivo era escribir de una manera práctica, con los pies en la tierra, acerca de lo que el marco de trabajo *Scrum* hace posible. Esperamos que sea fácil de leer, aclare posibles confusiones y ayude a profundizar en su comprensión.

Hemos escrito este *whitepaper* con Johannes Schartau para nuestro libro «*The Zombie Scrum Survival Guide*».

# Un Poco de Historia



El [marco de trabajo Scrum](#) fue desarrollado por Ken Schwaber y Jeff Sutherland en la década de 1990. Se formalizó en 1995 para abordar la complejidad inherente al desarrollo de productos y de *software*. Recientemente, el marco de trabajo *Scrum* está siendo aplicado con éxito a problemas complejos en una amplia variedad de áreas. Desde el marketing al cambio organizacional. Y de la investigación científica al desarrollo de *software*. El marco de trabajo *Scrum*, dondequiera que lo uses, está construido sobre tres pilares que permiten el control del proceso empírico.

## Los Pilares del Empirismo

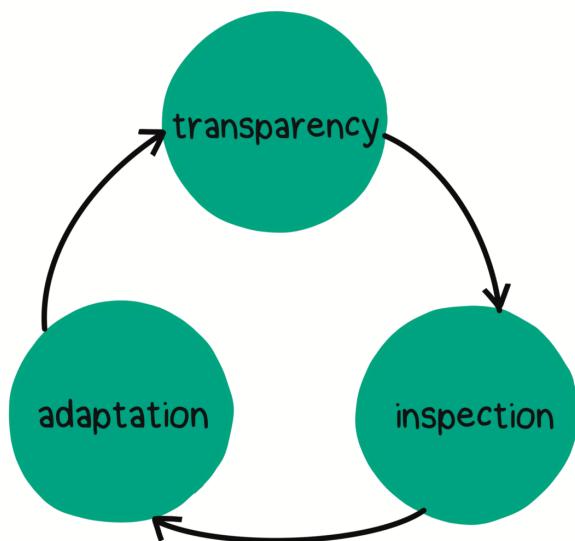


El marco de trabajo *Scrum* está construido sobre tres pilares que permiten el control del proceso empírico:

- **Transparencia:** recopilas datos, como métricas, *feedback*, y otras experiencias, para descubrir qué está ocurriendo;
- **Inspección:** inspeccionas el progreso con todos los involucrados y evalúas qué implicaciones tiene para vuestra visión del producto;
- **Adaptación:** realizas cambios que esperas que te acerquen más a tu visión;

Este ciclo se repite tan a menudo como sea necesario para detectar desviaciones, hallazgos inesperados y oportunidades potenciales que surgen mientras se realiza el trabajo. Esto no ocurre una vez al año o cuando se haya terminado el proyecto, sino de forma continua: diaria, semanal o mensualmente. En lugar de tomar decisiones basándonos en suposiciones sobre posibles futuros, tomamos decisiones basándonos en los datos que hemos recogido hasta el momento. Esto es el empirismo. Y en este artículo, descubrirás cómo todo dentro del marco de trabajo *Scrum* está diseñado en torno a estos pilares.

«En lugar de tomar decisiones basándonos en suposiciones sobre posibles futuros, tomamos decisiones basándonos en los datos que hemos recogido hasta el momento.»

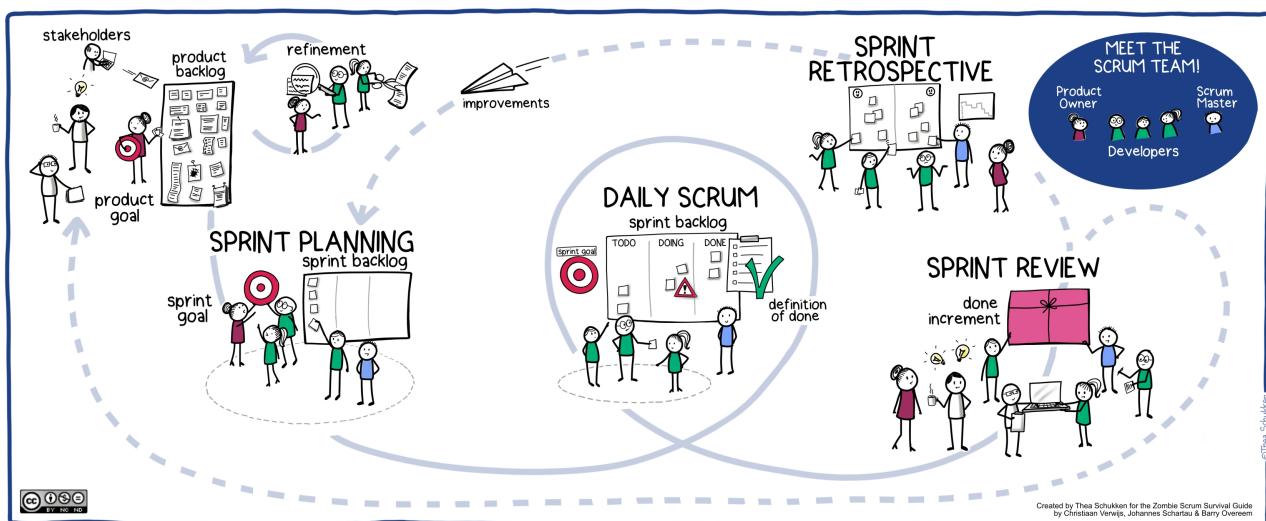


Created by Thea Schukken for the Zombie Scrum Survival Guide  
by Christiaan Verwijs, Johannes Scharbau & Barry Overeem



# Scrum: Un Marco de Trabajo para el Control del Proceso Empírico

Una cosa es decir que necesitas transparencia en el trabajo que haces para un producto, que necesitas inspeccionar ese trabajo frecuentemente y que los descubrimientos que emergen de dicha inspección guiarán la adaptación. Otra es poner todo eso en práctica, de manera tangible y concreta. Esto es lo que hace de *Scrum* un marco de trabajo; ofrece cinco eventos, que se repiten, para trabajar en tres artefactos, tres responsabilidades para dar apoyo, más varios principios y reglas que lo fusionan todo en un conjunto bien cohesionado.



## Los Artefactos de Scrum

El primer pilar del control del proceso empírico es la «Transparencia». Para poder inspeccionar el progreso de nuestro trabajo sobre un producto de forma frecuente y tomar decisiones sobre qué (más) se precisa, necesitamos tener algo que poder inspeccionar. Al hablar de hacer «transparente» el trabajo sobre un producto nos referimos a posibilitar que esté disponible de tal manera, que todas aquellas personas que tengan algún interés en él, puedan verlo, para validar suposiciones y para generar nuevas ideas.

El marco de trabajo *Scrum* requiere que los equipos sean transparentes, al menos, respecto a tres elementos de su trabajo sobre el producto. A estos elementos los llamamos artefactos en *Scrum*, y son los medios principales para recoger los datos y experiencias necesarios para la toma de decisiones de cara al futuro.

### El Product Backlog

El primer artefacto es el *Product Backlog*. Permite dar visibilidad a todo el trabajo necesario para alcanzar el *Product Goal*. El *Product Goal* permite encauzar el trabajo del *Product Backlog*, al facilitar que el Equipo *Scrum* se comprometa con un único objetivo a largo plazo, en relación al producto. A medida que los Equipos *Scrum* trabajan con el *Product Goal* en mente, el *Product Backlog* cambia para reflejar las nuevas ideas y oportunidades que surgen durante ese trabajo. Se debe establecer un nuevo *Product Goal* únicamente cuando el anterior se ha conseguido o abandonado, de otro modo la atención y el compromiso sufrirán.



Cada idea, pregunta, hipótesis, característica, defecto o cualquier otra tarea que el Equipo *Scrum* considere como necesaria para alcanzar el *Product Goal*, está representada en el *Product Backlog* como un elemento individual. Algunos elementos serán muy amplios y poco claros mientras que otros serán pequeños y específicos. Todos los elementos se ordenan de acuerdo con su relevancia en relación a la visión del producto y a sus *stakeholders*.

Un producto tiene un *Product Backlog* con un *Product Goal*, sin importar cuántos equipos estén trabajando en él. Si no, la transparencia y la atención sufren, ya que no hay una única «fuente de la verdad» sobre el trabajo que se necesita hacer y el orden en el que debe ser hecho.

## El Sprint Backlog

El segundo artefacto es el *Sprint Backlog*. Consiste en la selección de los elementos del *Product Backlog* que los Desarrolladores consideran necesarios para lograr el *Sprint Goal* al que se han comprometido para ese *Sprint*. El *Sprint Backlog* hace transparente todo el trabajo que el Equipo *Scrum* está realizando, o va a realizar, durante el *Sprint* actual. Cada Equipo *Scrum* tiene su propio *Sprint Backlog*, incluso cuando varios equipos están trabajando en un único producto. El *Sprint Backlog* no es estático y cambia conforme el equipo va aprendiendo durante el *Sprint*. Los Desarrolladores, en estrecha colaboración con el *Product Owner*, añaden o eliminan elementos basándose en cuánto tiempo queda en el *Sprint* la relevancia o necesidad de esos elementos para el *Sprint Goal*.

## El Incremento

El tercer artefacto es el Incremento. Cada Incremento representa un paso hacia la consecución del *Product Goal*. En un *Sprint*, se crea un Incremento cada vez que se completa un elemento del *Sprint Backlog* proveniente del *Product Backlog*. Para evitar confusión y variabilidad en las expectativas, un elemento se considera «hecho», solo cuando es verificado de forma exhaustiva y cumple con la Definición de Hecho acordada por el Equipo *Scrum*. Los Equipos *Scrum* pueden crear múltiples Incrementos a lo largo de un *Sprint*, incluso pueden entregarlos a los *stakeholders* antes de la *Sprint Review*.

El propósito de cada Incremento es validar las asunciones sobre el trabajo que se ha hecho hasta la fecha. Quienes tengan un interés en el producto pueden inspeccionarlo y determinar si cumple sus necesidades, si entienden cómo funciona y si satisface sus expectativas. El Incremento es un impulsor de nuevas ideas, dado que interactuar con algo tangible permite descubrir nuevas posibilidades.

El Incremento es el medio principal por el que los Equipos *Scrum* pueden ejercitar el control del proceso empírico aplicado a problemas complejos. Cada Incremento comienza con una suposición, una hipótesis o una idea potencialmente valiosa de cómo un producto puede acercarse a su objetivo. Esto se refleja en el *Sprint Goal*. El trabajo necesario para lograr ese objetivo se refleja en el *Sprint Backlog*.

Si múltiples Equipos *Scrum* trabajan en un único producto, todos ellos integrarán su trabajo en un único Incremento cada *Sprint*, de manera que pueda realizarse una inspección eficaz.

## ¿Qué Más?

Obviamente, hay muchos otros elementos en el trabajo de un equipo que se pueden hacer transparentes. Por ejemplo, puedes crear un mapa de *stakeholders* para comprender mejor quiénes son y dónde están. Puedes recoger métricas e inspeccionarlas frecuentemente. Estos artefactos están bien y merecen ser inspeccionados. Pero para la elaboración empírica de productos, el marco de trabajo *Scrum* recomienda que los equipos comiencen a practicar la transparencia con los tres artefactos fundamentales: el *Product Backlog*, el *Sprint Backlog* y el Incremento.



# Los Eventos de Scrum



Para poder tomar decisiones fundadas acerca de qué es lo siguiente que hay que hacer, necesitamos dar sentido, de forma frecuente, al *Product Backlog*, al *Sprint Backlog* y al Incremento, contando para ello con todos los interesados. En esto consiste la «Inspección». Y se presenta de muchas formas. Podemos dar sentido al *Product Backlog* mirándolo y sacando conclusiones – como «que es muy largo», «que no deberíamos hacer este elemento antes que este otro», o «¿cuándo podemos entregar esto a los *stakeholders*?». Podemos dar sentido al Incremento involucrando a los usuarios y validando nuestras suposiciones con ellos – como «¿los usuarios entienden esta nueva funcionalidad?», o «¿esta funcionalidad satisface su necesidad?». Y podemos dar sentido al *Sprint Backlog* asegurándonos de que es factible y que refleja nuestro plan para el *Sprint*.

Sea cual sea la forma en la que se presente, la inspección ofrece la mejor base para tomar decisiones cuando se fundamenta en algo tangible y concreto. En el caso del desarrollo de productos, inspeccionar una funcionalidad completa y desplegada es la mejor manera de realmente validar nuestras suposiciones sobre esa funcionalidad y cómo esta se usa. Ver una presentación o una propuesta de diseño de una funcionalidad puede parecer útil, sin embargo, todos los presentes todavía seguirán haciendo una gran cantidad de suposiciones (probablemente diferentes) sobre cómo funcionará cuando se lleve a implementar.

Con el objetivo de trabajar de forma empírica, el marco de trabajo *Scrum* propone que los equipos tengan al menos cinco «momentos», que se repiten, en los que la inspección tiene lugar. Cada uno de ellos tiene un límite máximo de tiempo (o *time-box*) y ofrece una perspectiva concreta del trabajo que se está haciendo. En su conjunto, ofrecen una imagen completa. Nos referimos a los cinco eventos de *Scrum*. Aunque se les menciona frecuentemente como «reuniones», no están pensados como reuniones en el sentido tradicional de la palabra, en las que un grupo de gente se sienta alrededor de una mesa, muertos de aburrimiento. De hecho, cuando los eventos de *Scrum* se llevan a cabo teniendo en cuenta su propósito, la necesidad de otras reuniones disminuye y se crean las condiciones adecuadas para que la colaboración y la coordinación fluyan de forma natural en los momentos precisos.

## El Sprint

El primer evento es la duración acotada (o *time-box*) del propio *Sprint*. De manera similar a cuando resuelves un rompecabezas complejo comenzando por un área pequeña, el propósito de cada *Sprint* es intentar resolver una parte de un problema complejo. Esta solución parcial está representada por una versión incremental del producto – el Incremento – que puede ser inspeccionada por todos para decidir qué debería pasar después. Este Incremento debería estar, como mínimo, en un estado que permita ser entregado a los *stakeholders*, después del *Sprint*, con el clásico «con solo pulsar un botón» si el *Product Owner* decide hacerlo. Sería incluso mejor poder ir entregando a lo largo del *Sprint*, acelerando el aprendizaje. Aunque el éxito de los *Sprints* variará (dado que incluso un único *Sprint* conlleva trabajo complejo e impredecible) se puede aprender mucho de este rompecabezas de todas formas.

«De manera similar a cuando resuelves un rompecabezas complejo comenzando por un área pequeña, el propósito de cada *Sprint* es intentar resolver una parte de un problema complejo.»

Los *Sprints* deberían ser siempre de la misma longitud con el objetivo de crear cadencia y predictibilidad para el equipo y sus *stakeholders*. Cuando un *Sprint* es demasiado largo, se pierden valiosas oportunidades para validar suposiciones y para asegurar que el trabajo progresá en la dirección adecuada. A medida que la longitud de los *Sprints* aumenta, también lo hace el riesgo de malgastar tiempo creando algo incorrecto. El marco de trabajo *Scrum* no especifica la longitud de los *Sprints*, solo que deben durar menos de un mes. Es decisión del equipo determinar cuán rápido quieren y pueden aprender. De forma general, los *Sprints* deberían ser lo más cortos posible mientras permitan que los Desarrolladores entreguen una nueva versión significativa del producto que se ajuste a la meta de un *Sprint*.



Mientras que el *Sprint* es una oportunidad con límite máximo de tiempo para explorar un área particular del problema complejo que supone el desarrollo de productos, los otros cuatro eventos representan oportunidades específicas dentro del *Sprint* que promueven la inspección y la adaptación.

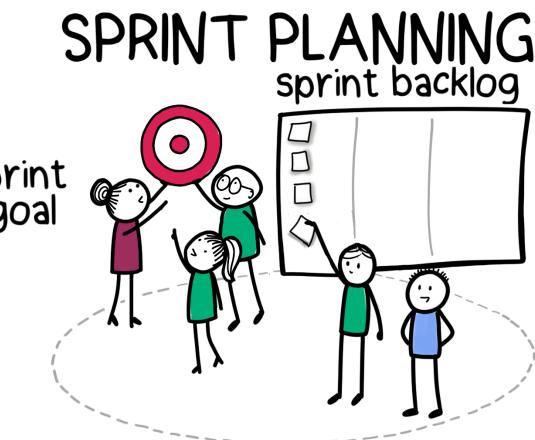


## Sprint Planning

Cada *Sprint* en *Scrum* comienza cuando el Equipo *Scrum* realiza un plan básico para dicho *Sprint*. A esto lo llamamos *Sprint Planning*. La primera cuestión, y la más importante, es cuál va a ser la meta (*goal*) para el *Sprint*. Sin una meta, no hay un propósito claro que anime a los miembros del equipo a unir esfuerzos durante el *Sprint*. Una meta para un *Sprint* puede ser entregar un conjunto coherente de funcionalidades que resuelva un problema particular. Puede ser satisfacer una necesidad de un grupo de *stakeholders*. Puede ser una hipótesis o suposición crítica que el *Product Owner* quiere verificar con el Incremento resultante del *Sprint*.

Aunque el *Product Owner* hace bien al asistir al *Sprint Planning* con un objetivo en mente, el Equipo *Scrum* al completo trabaja para refinar este objetivo de forma que se convierta en una meta valiosa y factible dentro de los límites temporales del *Sprint*. Los Desarrolladores trabajan con el *Product Owner* para seleccionar del *Product Backlog* el trabajo que debería llevarse a cabo durante el *Sprint* para lograr esa meta, reordenando el *Product Backlog* según sea necesario. Esta selección se convierte en el *Sprint Backlog*. Dado que los Desarrolladores son los encargados de hacer el trabajo, son los que poseen la última palabra acerca de qué se incluye en el *Sprint Backlog*.

Para *Sprints* de un mes, el *Sprint Planning* no debe durar más de ocho horas. Cuanto más corto sea el *Sprint*, menos tiempo debe durar. El *Sprint Planning* se completa cuando hay un esquema aproximado para el *Sprint* y un plan más detallado para el primer par de días. Este plan suele reflejarse en un desglose del trabajo para los primeros días del *Sprint*. A medida que el *Sprint* progresá, los Desarrolladores traducen ese plan aproximado —de acuerdo con el *Sprint Goal* y el *Sprint Backlog*— en un plan más concreto sobre cómo colaborar para llevarlo a cabo. Al menos una de las oportunidades para concretar el plan inicial es la *Daily Scrum*.



Created by Thera Schukken for the Zentrale Scrum Grundkurse  
by Christian Völkel, Johannes Scherfke & Barry Overman

## Daily Scrum

La *Daily Scrum* tiene lugar cada 24 horas y permite a los Desarrolladores sortear la complejidad que cualquier *Sprint* lleva aparejada. Evalúan el progreso de su trabajo hacia el *Sprint Goal*, progreso que se muestra de manera transparente en el *Sprint Backlog*, y se hacen los ajustes oportunos. Por ejemplo, pueden encontrarse problemas inesperados que requieran una colaboración estrecha o pueden descubrir que necesitan añadir más trabajo al *Sprint Backlog* para lograr el *Sprint Goal*. Pueden encontrar impedimentos; problemas que están bloqueando el logro del *Sprint Goal* y que están fuera de su alcance.

La *Daily Scrum* no debería durar más de 15 minutos. Es una reunión corta y de mínimos, en la que el equipo coordina el trabajo colaborativo para las siguientes 24 horas. Si más coordinación fuese necesaria, los Desarrolladores pueden llevarla a cabo durante el resto del día. El marco de trabajo *Scrum* no prescribe cómo hacer una *Daily Scrum* de forma efectiva, sino que anima a los Desarrolladores a encontrar la receta que mejor funcione para ellos.

## DAILY SCRUM

sprint backlog



Created by Thera Schukken for the Zentrale Scrum Grundkurse  
by Christian Völkel, Johannes Scherfke & Barry Overman



## Sprint Review

La *Sprint Review* tiene lugar al final del *Sprint*, antes de la *Sprint Retrospective*. Su propósito es inspeccionar el trabajo que se ha hecho hasta la fecha y decidir cuáles son los siguientes pasos, en base a lo aprendido. La *Sprint Review* es, al menos, un momento en el que la gente que construye el producto y la gente que está interesada en él se reúnen e inspeccionan los resultados del *Sprint*. Teniendo en cuenta también las condiciones del mercado, los cambios organizacionales, el presupuesto y el cronograma, se deciden de manera conjunta los siguientes pasos.

La *Sprint Review* no debe durar más de cuatro horas para un *Sprint* de un mes. Cuanto más corto sea el *Sprint*, más corta tiende a ser la *Sprint Review*.

El resultado de la *Sprint Review* son una serie de ajustes del *Product Backlog* basados en lo aprendido. Esto puede suponer nuevas ideas que surjan durante la *Sprint Review*, fallos detectados, cambios en elementos existentes en el *Product Backlog* o la repriorización del propio *Product Backlog*. En cierto sentido, la *Sprint Review* gira en torno a la pregunta: «basándonos en lo que hemos aprendido en este *Sprint*, ¿cuáles son los siguientes pasos a seguir?». Su respuesta nos proporciona una información muy valiosa para el *Sprint Planning* y futuros *Sprint Goals*.

Realizar solo una presentación durante la *Sprint Review* sobre lo que han hecho los Desarrolladores no implica «inspección». De hecho, inspeccionar el incremento significa probarlo juntos y dar *feedback*. Preferimos describir las *Sprint Reviews* como «fiestas del *feedback*» en lugar de «demos». Las *Sprint Reviews* no deberían ser la primera ocasión en la que el *Product Owner* ve qué han hecho los Desarrolladores. La *Sprint Review* es en realidad el momento en el que el *Product Owner* y los *stakeholders*, que acuden invitados por este, inspeccionan el producto juntos.

«Realizar solo una presentación durante la *Sprint Review* sobre lo que han hecho los Desarrolladores no implica "inspección".»

## SPRINT REVIEW



Created by Thies Schüken for the Zombie Scrum Survival Guide  
by Christian Verwoerd, Johannes Schatz & Barry Overeem

## Sprint Retrospective

La *Sprint Retrospective* ocurre al final del *Sprint*, normalmente después de la *Sprint Review*. Participa el Equipo *Scrum* al completo. Su propósito es inspeccionar cómo trabajó el Equipo *Scrum* de manera conjunta para lograr el *Sprint Goal*, e identificar acciones concretas que puedan ser implementadas en el siguiente *Sprint* con el objeto de incrementar la eficacia y la calidad. Mientras que la *Sprint Review* está más enfocada hacia el producto y el contenido del trabajo que se ha hecho, la *Sprint Retrospective* se dirige a inspeccionar el proceso de cómo se ha hecho el trabajo.

Para *Sprints* de un mes, la *Sprint Retrospective* no debe durar más de tres horas. Pero cuanto más corto es el *Sprint*, más corta tiende a ser su duración. El Equipo *Scrum* puede hacer ajustes para trabajar de forma más efectiva, basándose en lo que ha aprendido, tras inspeccionar su colaboración. Esto puede implicar una actualización de la Definición de Hecho (*Definition of Done*), investigar nuevas herramientas o tecnologías, un cambio en los acuerdos de trabajo o en la composición del equipo. Al menos una acción de mejora viable ha de ir directamente al *Sprint Backlog* del siguiente *Sprint*.



Created by Thea Schukken for the Zombie Scrum Survival Guide  
by Christiaan Verwijs, Johannes Schartau & Barry Overeem



## El Refinamiento del Product Backlog

Entonces, ¿qué pasa con el refinamiento del *Product Backlog*? La Guía de *Scrum* lo menciona como algo que tiene que ocurrir en algún momento. ¿No debería ser un evento? Si bien es una parte esencial del marco de trabajo *Scrum*, no es un evento como los otros. Más bien, es una actividad continua. Y aunque esto pueda parecer un juego de palabras, contiene una diferencia importante. Empecemos conociendo cuántas formas de refinamiento existen:

- Aclarar elementos en el *Product Backlog* que son demasiado confusos como para empezar a trabajar con ellos. Esto se hace preferiblemente con la gente para la que estás construyendo los elementos (los *stakeholders*);
- Descomponer los elementos del *Product Backlog* que por su tamaño no se pueden completar en un *Sprint* (lo que generalmente también significa que son poco claros);
- Re-ordenar el trabajo en el *Product Backlog* según sea necesario para hacer los siguientes *Sprints* tan tranquilos y valiosos como sea posible;
- Añadir o eliminar elementos del *Product Backlog* a medida que surgen nuevas ideas;
- Estimar el esfuerzo que conlleva implementar ciertos elementos. Esto no tiene por qué ser tan «formal» como asignar *story points* (una práctica opcional en *Scrum*), tallas de camiseta o cualquier otra técnica de medida que se use. Una coronada («sí, sabemos lo que hay que hacer, y creemos que se puede hacer en un *Sprint*») también es válida;

En cierto sentido, los elementos en un *Product Backlog* son recordatorios de «conversaciones que necesitamos tener en un futuro». Y el refinamiento es el proceso continuo de llevar a cabo esas conversaciones. Algunas veces esto significa hablar con los *stakeholders* sobre un elemento que puede acabar en el siguiente *Sprint*, mientras que otras puede ser clarificar un elemento en el que el equipo ya está trabajando. Algunas de esas conversaciones requerirán al equipo completo y otras a pequeños grupos. Algunos refinamientos pueden hacerse incluso individualmente.

Por lo tanto, en lugar de ver el refinamiento del *Product Backlog* como una reunión formal, que ocurre una vez en el *Sprint*, en la que participa todo el equipo, debería ser una serie de actividades recurrentes para refinar el trabajo del *Product Backlog* para los siguientes *Sprints*. Es una actividad continua que ocurre de diferentes formas, en la que participan diferentes configuraciones de personas. Solo el Equipo *Scrum* puede decidir cómo llevarlo a cabo.



# Las Responsabilidades de Scrum

La colaboración entre profesionales competentes es esencial para resolver problemas complejos. Las soluciones creativas e innovadoras surgen con más facilidad cuando reúnes las perspectivas de profesionales competentes. Para facilitar esta colaboración y reducir la complejidad en la comunicación y toma de decisiones, el marco de trabajo *Scrum* limita, de manera intencionada, las responsabilidades (o «roles») a tres. En lugar de roles en sentido jerárquico, donde uno tiene autoridad sobre los demás, cada uno de ellos representa una perspectiva diferente que debe incluirse para trabajar empíricamente. Los tres juntos son suficientes para trabajar empíricamente en cualquier entorno. Ninguna otra responsabilidad o rol es necesario (y probablemente solo serían un estorbo).

## MEET THE SCRUM TEAM!

Product Owner



Scrum Master



Developers



Created by Thea Schukken for the Zombie Scrum Survival Guide  
by Christiaan Verwijs, Johannes Schartau & Barry Overeem



## **El Product Owner**

El *Product Owner* incluye la perspectiva del qué es valioso (y qué no lo es) en la visión del producto. A medida que el Equipo *Scrum* invierte tiempo y dinero trabajando en el producto, el *Product Owner* se asegura de que esa inversión devolverá valor a los *stakeholders*. Es importante una colaboración estrecha con las personas que tienen interés en el producto, así como con los Desarrolladores, para decidir qué es valioso y qué no.

«El *Product Owner* incluye la perspectiva del qué es valioso (y qué no lo es) en la visión del producto.»

Un producto tiene un *Product Owner* y un *Product Backlog* con un *Product Goal*. Para mantener alta la velocidad en la toma de decisiones y que la adaptación pueda realizarse rápidamente, es necesario que los *Product Owners* tengan la máxima autoridad con respecto al producto. Ellos tienen la última palabra sobre la visión del producto, el contenido del *Product Backlog* y cómo gastar el presupuesto (o incluso establecerlo).

El *Product Owner* se asegura de que hay un *Product Goal*, que hay un *Product Backlog* ordenado, y que ambos están disponibles tanto para el Equipo *Scrum* como para los *stakeholders*. Esto no quiere decir que el *Product Owner* sea la única persona en el Equipo *Scrum* encargada de hacer este trabajo. A fin de maximizar el valor del trabajo hecho por los Desarrolladores en cada *Sprint*, tiene sentido que el *Product Owner* trabaje de forma activa con estos para escribir elementos, refinarlos y ordenarlos. Es todo cuestión de colaboración.

## **Los Desarrolladores**

Los Desarrolladores son todos los miembros del Equipo *Scrum* que contribuyen activamente a la entrega de un Incremento, logrando así el *Sprint Goal*. No importa qué tipo de trabajo realizan o cuál es la denominación formal de los puestos, para el marco de trabajo *Scrum* son todos «Desarrolladores». Todos juntos aportan la perspectiva de cómo hacer el trabajo necesario para materializar la visión de un producto y mantener alta su calidad.

Debido a que el desarrollo de productos es un trabajo complejo, incluso el futuro de un único *Sprint* es difícil de predecir. Es muy probable que surjan cuestiones, desafíos y problemas que impidan al equipo entregar ese Incremento. También es probable que surjan nuevas ideas durante el *Sprint* sobre lo que se debe incluir o dejar fuera del Incremento. Esta naturaleza impredecible de los *Sprints* determina tres requisitos de vital importancia en cuanto a cómo trabajan y se organizan los Desarrolladores.

«Los Desarrolladores aportan la perspectiva del cómo hacer el trabajo necesario para lograr alcanzar la visión del producto y mantener alta su calidad.»

El primer requisito es trabajar duro para minimizar sus dependencias con otras personas, departamentos y habilidades que no son parte de su equipo, pero que son frecuentemente necesarias para entregar un incremento considerado como hecho. Cada dependencia es algo sobre lo que tienen un control limitado y puede frenar o bloquear completamente su capacidad de entregar un Incremento hecho cada *Sprint*. Esto afecta negativamente a su capacidad para trabajar empíricamente. Las dependencias se pueden minimizar de diferentes formas, con la automatización (por ejemplo, para el despliegue y el testeо), el entrenamiento de nuevas habilidades o la incorporación de gente con las habilidades necesarias al Equipo *Scrum*.

El segundo requisito es que los Desarrolladores reduzcan los cuellos de botella derivados de cómo se distribuyen las habilidades dentro del equipo. Tener un tester en un Equipo *Scrum* es mejor que no tener ninguno en absoluto. Pero si el tester está desbordado por la cantidad de trabajo, todo el equipo ve mermada su velocidad. Lo mismo ocurre con otras habilidades, como el desarrollo de *back-end*, el diseño y el análisis. El equipo trabaja duro para asegurarse de que uno o más de sus miembros sean capaces de llevar a cabo trabajos que normalmente no realizan. El objetivo aquí no es que todos sean capaces de hacer todos los trabajos con el mismo nivel de maestría, esto es obviamente poco realista e irrespetuoso con las habilidades que la gente lleva años desarrollando, sino asegurarse de que otros pueden ayudar o asumir otros trabajos según sea necesario. El marco de trabajo *Scrum* sintetiza esta idea afirmando que «Los Equipos *Scrum* son multifuncionales».



El tercer requisito es que los Desarrolladores, a menudo, deben confiar en su inteligencia y creatividad para superar los muchos acontecimientos inesperados que están destinados a suceder durante un *Sprint*. Por eso los Equipos *Scrum* deben autogestionar, en la medida de lo posible, decisiones sobre cómo trabajar y qué mejorar en cómo se hace ese trabajo. En equipos autogestionados, no hay un «jefe» designado que tome las decisiones por el equipo. Todas las personas que hacen el trabajo (es decir, los Desarrolladores) trabajan conjuntamente para tomar decisiones dentro de los límites del marco de trabajo *Scrum*.

## El Scrum Master

El *Scrum Master* aporta la perspectiva del control del proceso empírico y de la calidad, mediante los cuales la transparencia, inspección y adaptación toman forma dentro y fuera del Equipo *Scrum*. El *Scrum Master* está ahí para hacer que los elementos del marco de trabajo *Scrum* cobren vida en el equipo y de forma más amplia en la organización. Para conseguir esto, los *Scrum Masters* adoptan diferentes papeles, dependiendo de la situación en la que se encuentren:

- Maestro: enseña y explica el propósito del marco de trabajo *Scrum* como un medio para trabajar empíricamente. Trabaja duro para asegurarse de que todos comprendan cómo los artefactos, eventos, roles y principios promueven el empirismo y la agilidad;
- Facilitador: facilita los pilares del marco de trabajo *Scrum* para maximizar las oportunidades de transparencia, inspección y adaptación. Un ejemplo de esto es la facilitación de eventos *Scrum*, cuando el Equipo *Scrum* los solicite o requiera. Otro ejemplo es ayudar al *Product Owner* a encontrar y usar técnicas para gestionar el *Product Backlog* y a los *stakeholders*;
- Eliminador de Impedimentos: el *Scrum Master* elimina (o ayuda a eliminar) obstáculos que impiden que el Equipo *Scrum* logre sus *Sprint Goals*. El *Scrum Master* ayuda al Equipo *Scrum* a aumentar su capacidad de resolver problemas de manera autónoma. Esto es algo que los equipos tienen que aprender, y el *Scrum Master* les ayuda a hacerlo. Lo que puede considerarse un impedimento durante el primer *Sprint*, puede haberse convertido en un problema que el equipo puede resolver fácilmente por sí mismo durante un *Sprint* futuro;
- Agente de Cambio: ayuda a eliminar impedimentos que bloquean el empirismo y la agilidad en los entornos de trabajo de los Equipos *Scrum*. Por ejemplo, algunas organizaciones separan los «testeos» y «despliegues» en diferentes equipos o departamentos. O prácticas de recursos humanos que recompensan a los individuos, no a los equipos. La eliminación de estos impedimentos no es algo que un *Scrum Master* pueda hacer solo, por lo que generalmente trabajará junto a otros *Scrum Masters*, *Product Owners*, etc;
- Entrenador y Mentor: entrena al Equipo *Scrum* realizando poderosas preguntas abiertas. Son mentores de otros *Scrum Masters* y ayudan a los miembros del Equipo *Scrum* a encontrar mentores que tengan la experiencia y las habilidades para ayudarlos;

Los *Scrum Masters* son verdaderos líderes. En lugar de dirigir la atención hacia sí mismos, ayudan a otros a ser tan efectivos como sea posible. No gestionan ni dirigen al equipo diciéndoles qué hacer o cómo hacerlo. Solo hay un escenario en el que los *Scrum Masters* deben mantener una posición firme, y es aquel en el que las decisiones que se toman impactan negativamente en el proceso empírico o en el grado en el que los miembros del equipo se sienten seguros para asumir riesgos interpersonales.

«El *Scrum Master* aporta la perspectiva del control del proceso empírico y de la calidad, mediante los cuales la transparencia, inspección y adaptación toman forma dentro y fuera del Equipo *Scrum*.»



## Juntos, el Equipo Scrum

Juntas, las tres responsabilidades representan un Equipo *Scrum*. En conjunto tienen plena autoridad para tomar todas las decisiones relevantes para su producto. El *Product Owner* decide cómo usar el presupuesto disponible para maximizar el valor que se entrega a los *stakeholders*. Los Desarrolladores deciden cómo construir ese producto y tienen todas las habilidades necesarias para hacerlo. Y el *Scrum Master* garantiza que esto se haga de manera que se maximice el empirismo. No hacen falta más roles.

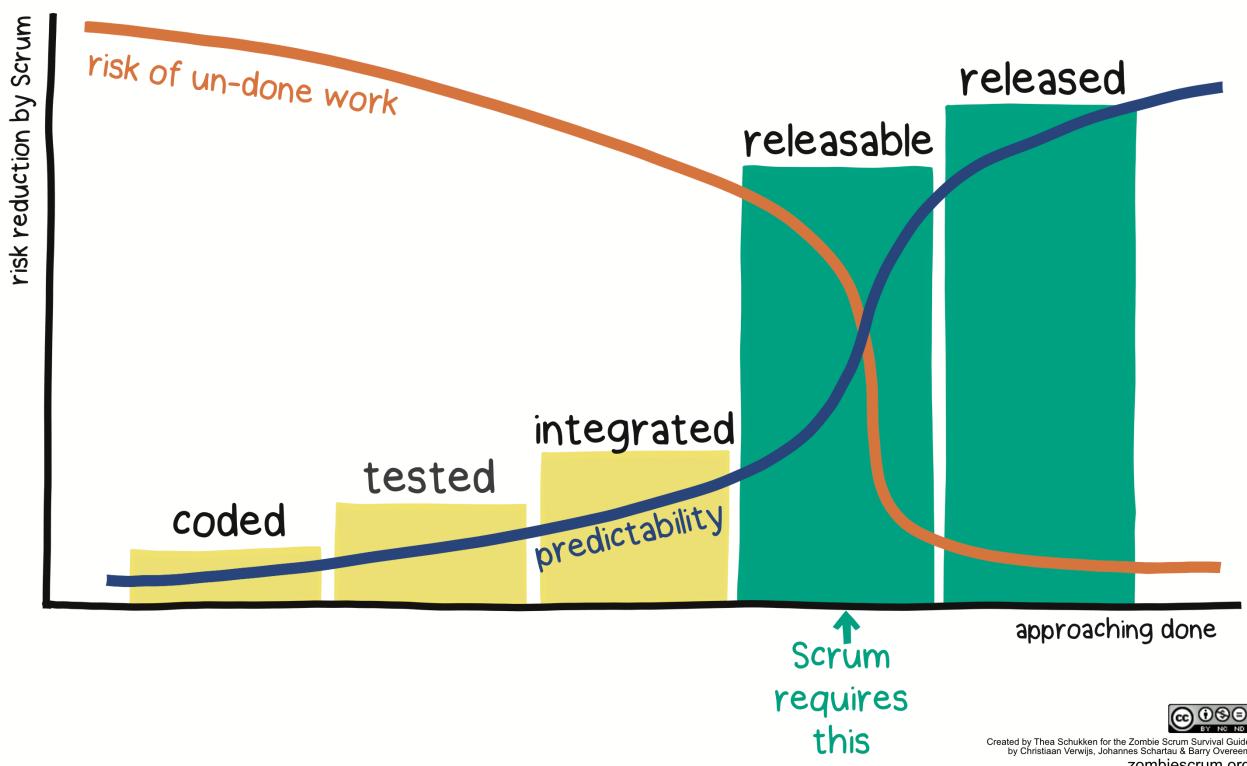
El marco de trabajo *Scrum* está diseñado de una manera que esta cita de Gunther Verheyen plasma con belleza: «Frente a la complejidad, la simplicidad es nuestro camino». Puede ser tentador agregar roles, reglas, prácticas y estructura al marco de trabajo *Scrum*, porque crea que su organización es diferente. Y aunque algunas situaciones pueden justificarlo, preservar la firmeza en mantener las cosas tan simples como sea posible es la mejor manera de avanzar en entornos complejos. Esto también se aplica al escalado. Agregar más equipos añade complejidad. En su lugar, explore formas de hacer más con menos gente.



# Dos Principios Rectores

Cuando la gente describe el marco de trabajo *Scrum*, a menudo lo hace explicando los eventos y roles. Aunque estos elementos son ciertamente importantes, su efectividad está determinada por lo bien que comprendan los equipos los dos principios rectores.

## Entregar un Incremento Hecho Cada Sprint



Si tuvieras que plasmar el propósito del marco de trabajo *Scrum* en una sola frase, este sería trabajar empíricamente entregando al menos un Incremento Hecho cada *Sprint*. Esta también es la razón por la que es tan importante que los Equipo Scrum dediquen tiempo a determinar qué hace falta para considerar un Incremento como Hecho. ¿Qué trabajo requiere esto? ¿Qué controles y pruebas son necesarios para cumplir con nuestras pautas de calidad internas? ¿Quién necesita estar involucrado? Este entendimiento común se denomina Definición de Hecho, y generalmente se traduce en una lista de verificación.

El trabajo en el *Product Backlog* tiende a estar repleto de suposiciones hasta que se completa y se pone en manos de los usuarios. Por ejemplo, «¿implementar este elemento va a mejorar la experiencia de los usuarios?», «¿entenderán cómo funciona?», «¿funciona lo suficientemente bien?» Habrá otras suposiciones sobre el trabajo que debe hacerse para implementar cierto elemento, tales como «¿cómo de fácil será probar y desplegar este elemento?», «¿qué problemas nos encontraremos cuando trabajemos en este elemento?», «¿comprendemos bien qué crear?».

Cada suposición representa un riesgo. El riesgo de que malinterpretes lo que ha pedido un usuario y tener que rehacer el trabajo. El riesgo de encontrarte con limitaciones técnicas cuando comienzas a escribir el código. El riesgo de que la funcionalidad tenga un rendimiento mucho peor de lo que se esperaba. El riesgo de que la escritura de tests automatizados resulte ser mucho más difícil. O el riesgo de gastar dinero y tiempo en una funcionalidad que acabe no siendo utilizada. Lo que todos estos riesgos comparten es que suponen un aumento inesperado del trabajo en algún momento. Y este tipo de trabajo «arrastrado» tiende a ser invisible hasta que aparece cuando menos te lo esperas, arruinando lo que estés haciendo en ese *Sprint*. Esto se llama «trabajo no hecho».

La mejor manera de prevenir el riesgo del trabajo no hecho es asegurarse de que cada *Sprint* culmine en, al menos, un Incremento Hecho que pueda ser potencialmente desplegado. Eso significa que el incremento ha sido completamente testeado y está funcionando. Los textos y las imágenes son los definitivos. La documentación y los paquetes de despliegue están actualizados. Y el rendimiento y la seguridad se ajustan a los estándares de la organización. Llegados a este punto, el incremento se puede poner a disposición de los usuarios con el consabido «con solo pulsar un botón». A partir de este momento, ya se sabe que habrá poco o ningún trabajo potencial sin terminar que pueda estropear el enfoque y la previsibilidad de futuros *Sprints*. Y lo que es más importante, al desplegar el producto puedes comprobar de manera empírica si el incremento realmente acerca el producto a la visión, en vez de asumir que lo hará.

«Conforme la capacidad de los Equipo *Scrum* para entregar Incrementos Hechos se mueve hacia la derecha, disminuye el riesgo de trabajo no hecho. Y aumenta la previsibilidad. Inspirado en el trabajo de Gunther Verheyen.»

Crear un Incremento Hecho cada *Sprint* es realmente un desafío. Pero es así a propósito. Porque poner tal cantidad de presión en el sistema que crea el producto (hacer todo lo necesario para conseguirlo) mostrará de forma clara dónde son necesarias las mejoras. Dónde faltan habilidades, herramientas y tecnologías. Dónde está estorbando la burocracia. Dónde se necesita eliminar impedimentos para poder trabajar, de verdad, empíricamente y reducir el riesgo que supone el trabajo complejo. Al mantener nuestro punto de vista en la entrega de un Incremento Hecho al menos una vez cada *Sprint*, el sistema comenzará a resentirse en los lugares correctos creando así oportunidades para la inspección y la adaptación.

«Al mantener el enfoque en la entrega de un Incremento Hecho al menos una vez cada *Sprint*, el sistema comenzará a resentirse en los lugares correctos, creando así oportunidades para la inspección y la adaptación.»

## Usar un Product Goal Común y un Sprint Goal para Crear Cohesión

La Guía de *Scrum* menciona la palabra «*goal*» 40 veces, más que cualquier otro elemento del marco de trabajo *Scrum*. Esto no significa que tenga que hacerse una interpretación radical en ese sentido, a la hora de tomar decisiones acerca de cómo trabajar con *Scrum*. Pero sí debería decirnos algo sobre la importancia que tienen los objetivos compartidos.

La razón por la cual el *Product Goal* y el *Sprint Goal* son tan importantes se remonta a la razón por la que surgió el marco de trabajo *Scrum*: para sortear de forma más efectiva los problemas complejos y reducir el riesgo. Los *Goals* nos ayudan con esta complejidad de tres maneras complementarias.



La primera es que ofrece una guía al Equipo *Scrum* cuando este tiene que decidir en qué emplear su tiempo. Algunas tareas del *Sprint Backlog* pueden ser más importantes que otras para conseguir el *Sprint Goal*. La segunda es que permite al equipo concentrarse en lo que es importante. Cuando las cosas se ponen difíciles, como suele suceder, el equipo puede decidir dejar a un lado cierto trabajo y priorizar otro. O, cuando el tiempo lo permite, durante el *Sprint* pueden agregar trabajo que descubren que es necesario para lograr el objetivo. Y tercero, los *Sprint Goals* promueven la colaboración al brindar al Equipo *Scrum* un propósito claro y común para auto-organizarse, en lugar de trabajar de forma individual. La colaboración hace posible el tipo de pensamiento divergente (o lateral) y el espíritu de equipo que se necesita para resolver problemas complejos.

Los *Product* y *Sprint Goals* también dan color y propósito a los diferentes eventos de *Scrum*, haciendo de ellos mucho más que simples reuniones de trabajo. El *Sprint Planning* gira en torno a la definición de la meta para el siguiente *Sprint* y a la selección del trabajo que se necesite para llevarla a cabo. La *Daily Scrum* se centra en cómo trabajarán los Desarrolladores las siguientes 24 horas para lograr esta meta, y qué impedimentos pueden estar bloqueando su capacidad para lograrlo. La *Sprint Review* trata de verificar el resultado del *Sprint Goal* con los *stakeholders*, de relacionarlo con el *Product Goal*, y de trabajar con ellos para identificar las próximas metas en las que centrarse. La *Sprint Retrospective* se centra en descubrir formas de trabajar juntos de manera más efectiva para lograr los *Sprint Goals*.

Pero el *Product Goal* y el *Sprint Goal* también ayudan a enfocar las tres responsabilidades del marco de trabajo *Scrum*. Guían y orientan a los Desarrolladores a la hora de autogestionarse. Permiten a los *Product Owners* concentrarse en la visión del producto y cómo traducirla en objetivos para varios *Sprints*, en lugar de involucrarse en los detalles de cómo se van a implementar. Y finalmente, los *Scrum Masters* pueden aumentar la efectividad del control del proceso empírico usando el *Product Goal* y el *Sprint Goal* como al canario en la mina de carbón. Que un equipo tenga problemas para crear los *Sprint Goals* es un buen motivo para que los *Scrum Masters* se pongan sus sombreros de Sherlock y comiencen a investigar para descubrir la causa. Los equipos pueden ser demasiado grandes o demasiado pequeños, pueden carecer de ciertas habilidades o personas. El *Product Owner* puede no tener autoridad o visión. O el refinamiento puede no estar teniendo lugar.

Esta explicación debería dejar bien claro por qué debería haber solo un *Product Goal* por producto y un *Sprint Goal* por *Sprint*. Múltiples objetivos solo introducen confusión en el enfoque, disminuyen el compromiso y limitan la transparencia. Por supuesto, se puede formular una nueva meta siempre que se alcance o se abandone la anterior.

Para resumir, en trabajos complejos, objetivos compartidos como los *Product Goals* y los *Sprint Goals*, son faros que te ayudan a llegar al puerto a través de la densa niebla. Sin ellos, es probable que te pierdas o te quedes varado.



# Cinco Valores Fundamentales que Permiten el Empirismo

Hasta este punto, hemos cubierto las mecánicas y los principios del marco de trabajo *Scrum*. Aunque hacen maravillas para permitir el control del proceso empírico, por sí mismos, sin un comportamiento en el que apoyarse, no supondrían nada diferente. ¿Cómo de honesta puede ser la inspección durante una *Sprint Review* cuando los Equipos *Scrum* tienen miedo a hablar de forma abierta sobre los desafíos técnicos que ven, y en su lugar recurren al mensaje de que «todo va bien»?, ¿cómo de efectivo puede ser un Equipo *Scrum* cuando el *Product Owner* tiene miedo de declinar o rechazar elementos en el *Sprint Backlog* que no se ajustan al *Goal*?, ¿cuánto valor generará un Equipo *Scrum* que sigue distrayéndose con la tecnología más nueva y popular?, ¿cómo afecta el cambio de composición del equipo a la voluntad de comprometerse a trabajar como tal, cuando las decisiones acerca de esos cambios se toman sin el conocimiento o aprobación de los afectados?

Es difícil trabajar de forma empírica, especialmente en organizaciones que no están acostumbradas a ello. Con el fin de dar a los equipos y a sus *stakeholders* una guía en la toma de decisiones, el marco de trabajo *Scrum* enfatiza específicamente cinco valores rectores. Por cada decisión a la que se enfrenten, los equipos pueden preguntarse cómo esta afectará a esos cinco valores:

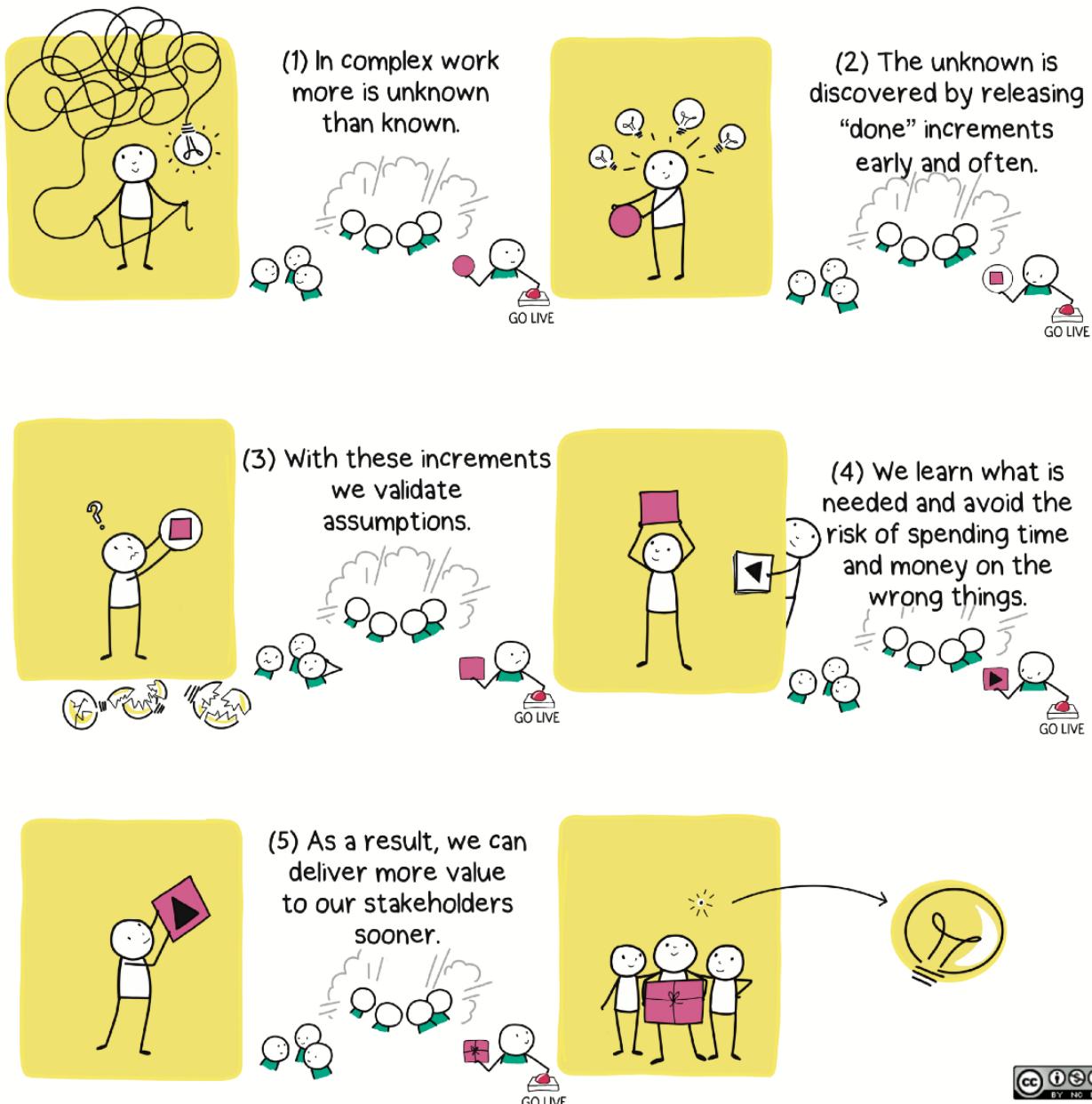
- Apertura: afrontar de forma abierta cómo van las cosas. ¿Qué está yendo bien?, ¿qué no?, ¿dónde están los desafíos y las oportunidades?
- Coraje: ser valiente para hacer lo correcto. Decir «no» a las cosas que impiden el proceso empírico. Mostrar coraje trabajando juntos en retos difíciles. Pedir y estar dispuesto a dar *feedback* sobre cosas de las que no se está seguro. Hacer preguntas y admitir lo que no sabes o aquello sobre lo que tienes dudas;
- Foco: mantenerse enfocado en el *Sprint Goal* y las metas del Equipo *Scrum*. Crear un espacio donde la gente pueda conservar y mantener el foco de atención;
- Respeto: respetar las habilidades, experiencia e inteligencia de los miembros del Equipo *Scrum*. Confiar en su habilidad para autoorganizarse en torno a problemas complejos. Y respetar la incertidumbre inherente al trabajo complejo;
- Compromiso: crear un entorno donde las personas puedan comprometerse personalmente a trabajar juntos como equipo hacia el *Sprint Goal*,

Aprender a integrar estos valores en nuestro comportamiento diario nos llevará toda la vida. Cuanto más competentes sean los equipos con ellos, más efectivas serán la transparencia, la inspección y la adaptación frecuentes. La buena noticia es que el marco de trabajo *Scrum* proporciona un excelente conjunto de límites para aprender y crecer en la adopción de estos valores.



# Todo Junto, un Marco de Trabajo Sencillo

Scrum es un marco de trabajo sencillo que permite abordar problemas complejos y adaptativos. Sólo prescribe lo que los equipos deben hacer para trabajar empíricamente, pero no cómo deberían hacerlo. Porque cada equipo, cada producto y cada organización es diferente, cada equipo debe encontrar la forma que mejor le funcione. Cuando lo consiguen, pueden reducir el riesgo del trabajo complejo, comenzar a entregar valor antes a sus *stakeholders* y ser más rápidos en reaccionar. Este viaje será más fácil para algunos Equipos Scrum que para otros. Pero el cambio siempre llega si persistes.



Created by Thea Schukken for the Zombie Scrum Survival Guide  
by Christiaan Verwijs, Johannes Schartau & Barry Overeem  
[zombiescrum.org](http://zombiescrum.org)

# Unleashing Teams All Over The World

We are The Liberators – Barry Overeem and Christiaan Verwijs. Our mission is to create data-driven products to unleash the superpowers of teams all over the world. We do this together with a growing community of patrons.

## **Awesome content for awesome teams**

We unleash teams with our [blogposts](#), our [podcast](#), our [newsletter](#), our [videos](#), and our frequent [meetups](#). While we offer most of this for free, we also have plenty of premium content in [our webshop](#) for you to explore.

## **Supported by the community**

We are super proud that The Liberators is almost entirely funded by the community. If you appreciate our work too, you can already support us for 12 dollars/year by becoming a [patron](#). In return, you gain free access to premium content, we share our work-in-progress and involve you in creating more awesome content.



## **Thank you for respecting our work!**

We work hard to create high-quality content that puts you in a position to unleash your team. We're sure you appreciate that a lot of our time and money goes into this. At the same time, content like this is our main source of income. It's what pays our bills :)

If you purchased this content, we ask only that you treat our work with respect and don't share it with people outside your team. If you stumbled on it elsewhere without paying for it, and it offers you value, would you consider [supporting us too](#)? You can also check out [our other offerings](#).