

SCRUM

An Agile Approach To Manage Successful Projects

2019

www.certmind.org

Scrum – An Agile Approach to Manage Successful Projects

Este libro fue desarrollado por CertMind como guía para la gestión exitosa de proyectos ágiles usando Scrum. El contenido aquí descrito, es el resultado de varios años de investigación realizada por nuestros expertos, estudiantes y voluntarios que aportaron su valioso conocimiento e hicieron posible la construcción de esta guía.

Aun cuando hoy en día existen varios marcos de referencia basados en Scrum, nuestro enfoque está centrado en el artículo “The New New Product Development Game” escrito en la Harvard Business Review por Hirotaka Takeuchi y Ikujiro Nonaka y posteriormente la guía desarrollada por Jeff Sutherland y Ken Schwaber.

Este libro es la base fundamental para la obtención de las certificaciones CM-SFC (Scrum Fundamentals Certified) – CM-SMC (Scrum Master Certified) – CM-SDC (Scrum Developer Certified) - CM-SPOC (Scrum Product Owner Certified) y CM Scrum Expert.

Contenido

Scrum – An Agile Approach to Manage Successful Projects	1
1. Introducción a Scrum	9
1.1 - Historia	9
1.2 - Definición de Scrum	10
1.3 - Anglicismos.....	10
1.4 - Estructura de la guía	11
1.5 - Usos de Scrum.....	11
1.5.1 - ¿Cuándo utilizar las metodologías ágiles?	12
1.6 - Aclaraciones de la guía	12
1.7 - Escalabilidad de Scrum	12

1.8 - ¿Scrum se implementa o se adopta?.....	13
2. El Equipo Scrum.....	13
2.1 - El Product Owner.....	14
2.2 - El Equipo de Desarrollo	15
2.2.1 - Tamaño del Equipo de Desarrollo	16
2.3 El Scrum Master (Scrum Master)	16
2.3.1 - El servicio del Scrum Master al Propietario del Producto (Product Owner).....	17
2.3.2 - El servicio del Scrum Master al Equipo de Desarrollo	17
2.3.3 - El servicio del Scrum Master a la Organización	18
2.4 - Los valores del equipo	18
2.4.1 – Compromiso	18
2.4.2 – Respeto.....	19
2.4.3 - Proactividad.....	19
2.4.4 – Receptividad - Apertura	19
2.4.5 – Foco.....	19
3. Eventos en Scrum.....	19
3.1 - El Sprint (Sprint)	19
3.1.1 - Consideraciones sobre la duración del Sprint	21
3.1.2 - Cancelación del Sprint.....	21
3.2 – Reunión de Planificación del Sprint.....	22

3.2.1 - Datos del evento	22
3.3 - Daily Scrum (Scrum Diario).....	23
3.3.1 - Datos del evento	23
3.4 – Reunión de Revisión del Sprint	24
3.4.1 Datos del evento	24
3.5 – Reunión de Retrospectiva del Sprint.....	25
3.5.1 - Datos del evento	25
4. Los Artefactos en Scrum.....	26
4.1 - Product Backlog	26
4.2 - Sprint Backlog.....	27
4.2.1 - Incremento.....	27
4.3 - Emisores de Información.....	28
4.3.1 - Scrum Board.....	28
4.3.2 - Burndown chart del Sprint.....	29
4.3.3 – Diagrama de flujo acumulado	30
4.4 – Registro de obstáculos/impedimentos	32
5. Principios de Scrum.....	32
5.1 - Control empírico de procesos (1).....	32
5.1.1 - Transparencia.....	32
5.1.2 - Inspección	33

5.1.3 - Adaptación.....	34
5.2 – Autoorganización (2)	34
5.2.1 - Colaboración de equipo	35
5.2.2 - Colaboración con el cliente.....	36
5.2.3 - Equipos multifuncionales.....	36
5.2.4 - Propiedad colectiva del producto	37
5.2.5 – Motivación del equipo.....	37
5.3 – Simplicidad (3).....	38
5.3.1 - Diseño simple	39
5.3.2 - Uso de herramientas de software.....	39
5.3.3 - ¿Vale la pena trabajar con procesos?.....	40
5.4 – Centrado en el valor para cliente (4).....	41
5.5 – Cumplimiento (5).....	41
5.5.1 - Bloques de tiempo.....	41
5.5.2 - Reglas del Equipo.....	42
5.6 – Desarrollo iterativo (6)	42
6. Consideraciones para la Gestión del Proyecto	43
6.1 – Contratación	43
6.1.1 - Roles del proyecto	43
6.1.2 – Contratación del equipo (Personal)	44

6.1.3	- Tipos de contratos	46
6.1.4	- Finanzas del proyecto	47
6.1.5	- Contratación de proveedores	49
6.1.6	- Los "Freelancers"	49
6.2	- Seguimiento y control del proyecto.....	50
6.2.1	- Oficina de Gestión de Proyectos Ágiles (APMO)	50
6.2.2	- Informes y reportes del proyecto	51
6.2.3	- Seguimiento de Calidad.....	53
6.3	- Gestión de Riesgos.....	54
6.3.1	- Ejemplos de riesgos comunes:.....	54
6.3.2	- Apetito de Riesgo	56
6.3.3	- Tolerancia al Riesgo	57
6.3.4	- Ciclo de vida de la gestión de riesgos	57
6.4	- Gestión de Cambios.....	61
6.5	- Diseño del producto.....	63
6.5.1	- Producto mínimo viable.....	63
6.5.2	- Análisis de viabilidad del producto	63
7.	El Ciclo de Vida en los Proyectos Ágiles.....	64
7.1	- Etapa 1: Inicio del proyecto.....	65
7.1.1	- Definir la visión del proyecto (1)	66

7.1.1.1	- Restricciones de un proyecto	66
7.1.1.2	- Identificar las partes interesadas.....	67
7.1.1.3	- Definición de terminado (DoD).....	67
7.1.2	- Formar el Equipo Scrum (2)	69
7.1.2.1	- Plan de colaboración.....	69
7.1.2.2	- Reunión de inicio de proyecto o Kickoff	69
7.1.2.3	- Modelo de desarrollo de equipos – Dr. Bruce Tuckman	70
7.1.2.4	- Equipos de alto rendimiento	73
7.1.3	- Construir el Product Backlog (3)	73
7.1.3.1	- Clasificando el Product Backlog	74
7.1.4	- Priorizar el Backlog (4)	74
7.1.4.1	- Priorización por urgencia.....	75
7.1.4.2	- Visual Story Mapping.....	76
7.1.5	- Definir el cronograma de entregas (5).....	76
7.1.5.1	- Calendario del equipo	77
7.1.6	- Definir la Arquitectura de Producto (6).....	78
7.1.6.1	- Sprint Cero	78
7.1.6.2	- Evaluación y Selección de Proveedores.....	79
7.2	- Etapa 2: Planificación	79
7.2.1	- Escribir las historias de usuario y tareas (7).....	80

7.2.1.1 - ¿Qué son las Historias de Usuario?	80
7.2.1.2 - Mockups y prototipos iniciales de proyecto.....	81
7.2.1.3 - Planificación en paralelo.....	81
7.2.1.4 - Desglosar las Historias de Usuario en tareas.....	82
7.2.2 - Priorizar las Historias de Usuario (8).....	82
7.2.3 - Reunión de Planificación del Sprint (9)	83
7.2.3.1 - Seleccionar el trabajo a desarrollar	83
7.2.3.2 - Estimación del trabajo seleccionado.....	83
7.2.3.3 - Compromiso del equipo.....	84
7.2.3.4 - ¿Cómo se conseguirá “terminar” el trabajo seleccionado?.....	85
7.3 - Etapa 3: Desarrollo del Sprint.....	86
7.3.1 - Desarrollar los entregables (10).....	87
7.3.1.1 - El ciclo de desarrollo para proyectos de software	87
7.3.2 - Scrum diario (11)	90
7.3.2.1 - Seguimiento del Progreso del Sprint.....	90
7.4 - Etapa 4: Revisión del Sprint.....	92
7.4.1 - Reunión de Revisión del Sprint (12)	92
7.4.1.1 - Deuda técnica	93
7.4.1.2 - Refinamiento del Product Backlog	93
7.4.2 - Reunión de Retrospectiva del Sprint (13).....	94

7.5	- Etapa 5: Implementación.....	95
7.5.1	- Planificación de la implementación (14).....	96
7.5.1.1	- Coordinación con operaciones	96
7.5.2	- Implementación de entregables (15)	96
7.5.2.1	- Confirmación de implementación exitosa	97
7.5.2.2	- Despliegues Fallidos.....	97
7.6	- Etapa 6: Cierre del proyecto.....	97
7.6.1	- Cierre del proyecto (16).....	98
7.6.2	- Reunión de Retrospectiva del proyecto (17).....	99
8	Scrum en Grandes Proyectos	100
8.1	- Distribución de equipos.....	100
8.1.1	- Program Owner	100
8.2	- Product Backlog en múltiples equipos	101
8.3	- Scrum de Scrums.....	101
8.4	- Coordinación de Sprints	102
8.5	- Guías de ajuste sobre las prácticas Scrum	103
9	Agradecimiento	104
10	Términos oficiales en inglés	104
10.	Lista de figuras.....	105
11.	Cambios respecto a la edición 2017.....	107

1. Introducción a Scrum

1.1 - Historia

La historia de Scrum se puede rastrear desde 1986 en un artículo de la Harvard Business Review, "The New New Product Development Game" escrito por Hirotaka Takeuchi y Ikujiro Nonaka, en el que analizaron el enfoque que utilizaban compañías como Fuji-Xerox, Canon, Honda, NEC, Epson, Brother, 3M, Xerox, y Hewlett-Packard para el desarrollo de sus productos.

Durante sus investigaciones, se dieron cuenta que dichas empresas compartían seis características: inestabilidad incorporada, equipos de proyectos autoorganizados, fases de desarrollo superpuestas, multi-aprendizaje, control sutil y transferencia organizativa de aprendizaje. Las seis piezas encajan como un rompecabezas, formando un proceso rápido y flexible para el desarrollo de nuevos productos. Igual de importante, este nuevo enfoque puede actuar como un agente de cambio: es un vehículo para introducir ideas y procesos creativos e impulsados por el mercado en una organización antigua y rígida.

Es también en este artículo donde se hace la comparación de los procesos de Scrum con el juego del Rugby, y de donde se origina su nombre Scrum.

El artículo original se puede leer en: <https://hbr.org/1986/01/the-new-new-product-development-game>

Posteriormente son Ken Schwaber y Jeff Sutherland quienes trabajaron en Scrum desde 1995, cuando conjuntamente lo presentaron en la conferencia OOPSLA en 1995. Esta presentación documentó principalmente el aprendizaje que Ken y Jeff habían obtenido a lo largo de los años e hicieron pública la primera definición formal de Scrum. Para hacer honor a los primeros lugares donde fue probado y perfeccionado, reconocemos a Individual, Inc., Newspaper, Fidelity Investments e IDX (en la actualidad GE Medical).

La Guía de Scrum documenta Scrum tal y como ha sido desarrollado, evolucionado, y mantenido por más de veinte años por Jeff Sutherland y Ken Schwaber. Otras fuentes proporcionan patrones, procesos e ideas que complementan al marco de trabajo Scrum. Estas pueden incrementar la productividad, valor, creatividad y satisfacción con los resultados.

1.2 - Definición de Scrum

Scrum es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor.

Scrum es:

- Ligero
- Simple de entender
- Difícil de dominar

Scrum es un marco de trabajo compuesto de prácticas que se han utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90.

Scrum no es un proceso, una técnica, o método definitivo. Todo lo contrario, es un marco de trabajo donde se pueden emplear un conjunto de diferentes procesos y técnicas. Scrum muestra la eficacia relativa de las técnicas de gestión de producto y de trabajo de modo que podamos continuamente mejorar el producto, el equipo y el entorno de trabajo.

1.3 - Anglicismos

En este libro se utilizan varios términos utilizados originalmente en inglés, esto con el fin de no causar confusiones con las distintas traducciones que a estos se les pueden dar. A continuación, encuentras el listado de términos que no fueron traducidos en este material:

- **Burndown chart:** Cuadro de incendio o avance
- **Framework:** Marco de trabajo
- **Product Backlog:** Lista de pendientes del producto
- **Product Owner:** Propietario del producto
- **Scrum Board:** Tablero Scrum
- **Scrum Master:** Maestro Scrum
- **Sprint:** Iteración
- **Sprint Backlog:** Lista de pendientes del Sprint

1.4 - Estructura de la guía

Esta guía se compone por:

- **Introducción a Scrum:** Donde se explican conceptos como: Los Equipos Scrum, los Eventos, los Artefactos y las reglas relacionadas.
- **Los principios de Scrum:** Donde se habla de 6 reglas esenciales para garantizar el éxito en la aplicación de Scrum.
- **Consideraciones para la Gestión del Proyecto:** Donde se explican los elementos clave que se deben tener en cuenta para garantizar el correcto desarrollo del proyecto.
- **El ciclo de vida del proyecto:** Esta sección es el corazón de la guía, y es aquí donde se explica cómo poner en marcha todos los conceptos aprendidos, describiendo las prácticas clave y su "orden lógico" en un proyecto de la vida real.

Cada componente dentro de la guía tiene un propósito específico y es esencial para el éxito en la adopción de Scrum.

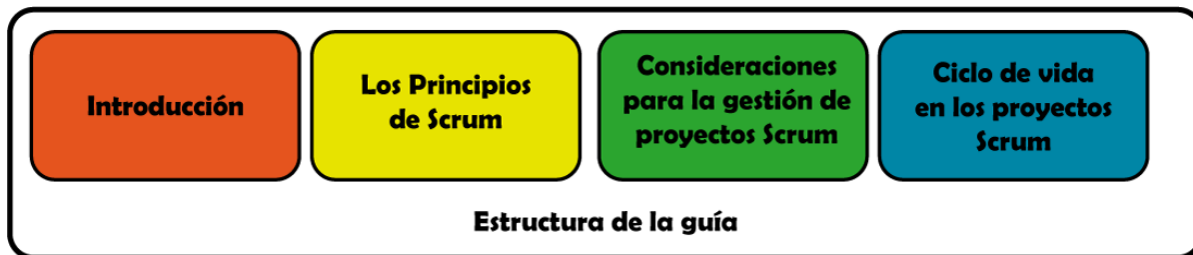


Ilustración 1 - Estructura de la guía

1.5 - Usos de Scrum

Scrum inicialmente fue desarrollado para gestionar y desarrollar productos. Sin embargo, su uso no se vio limitado a este campo, y con el tiempo la experiencia recaudada nos muestra que es útil para:

1. Investigar e identificar mercados viables, tecnologías, y capacidades.
2. Desarrollo de mejoras a productos ya existentes.
3. Desarrollo de productos que requieren lanzamientos diariamente o tantas veces como sea posible.
4. Desarrollo y mantenimiento en la Nube (online, seguridad, por-demanda) y otros entornos operacionales de desarrollo para el uso de productos.
5. Mantenimiento y renovación de productos.

6. Scrum se ha demostrado especialmente efectivo en la transferencia de conocimiento iterativamente e incrementalmente. Scrum es ampliamente utilizado para la construcción de productos y servicios.

1.5.1 - ¿Cuándo utilizar las metodologías ágiles?

Metodologías Cascada	VS	Metodologías Ágiles
<ul style="list-style-type: none"> • Proyectos de bajo riesgo • Proyectos con requisitos específicos definidos por el cliente (no negociables) 		<ul style="list-style-type: none"> • Proyectos de innovación en los que la investigación es clave. • Proyectos con alto grado de complejidad. • Probabilidad de que los requisitos cambien. • Entornos complicados, donde mantener la motivación es fundamental.

Ilustración 2 - Metodologías ágiles vs Metodologías Cascada

1.6 - Aclaraciones de la guía

- En el desarrollo de esta guía el término “Producto” se refiere a productos, servicios y cualquier otro entregable que pueda surgir como resultado de un ejercicio de proyecto.
- Cuando los conceptos “Desarrollo” y “Desarrollar” se utilizan en esta guía de Scrum, se refieren al desarrollo de las actividades propias del proyecto, no se ven limitados a desarrollo de software.

1.7 - Escalabilidad de Scrum

Scrum puede ser utilizado en grandes proyectos sin importar la cantidad de personas o actividades a desarrollar, esto siempre y cuando se respete la regla de formar pequeños equipos de personas, ya que este tipo de equipos son muy flexibles y adaptivos.

Las fortalezas de los equipos Scrum que permiten su fácil escalamiento son:

- Pueden operar individualmente, en varios, muchos, e incluso en redes de equipos.
- Su capacidad multifuncional les permite desarrollar, lanzar, operar y mantener el trabajo y el producto con la menor cantidad de dependencias posibles.
- La colaboración y la interacción continua entre equipos y con el cliente les permite desarrollar arquitecturas más sofisticadas en los entornos de desarrollo.

Ver más en la sección 8: *Scrum en Grandes Proyectos*.

1.8 - ¿Scrum se implementa o se adopta?

El marco de trabajo Scrum consiste más en un cambio cultural en toda la organización que en simplemente la implementación de las herramientas o técnicas aquí descritas, así que es erróneo pensar en la “implementación” de Scrum, lo más apropiado es hablar de la “Adopción” de Scrum, ya que se necesita poner en marcha las prácticas propuestas, realizar experimentos y estar ajustando continuamente el marco de trabajo a las necesidades cambiantes de la organización.

Para la adopción de las prácticas Scrum se requiere el apoyo de toda la organización, esta cultura orientada hacia lo ágil está basada en la simplicidad, el valor y el alto rendimiento que se necesita para la gestión de proyectos modernos.

2. El Equipo Scrum

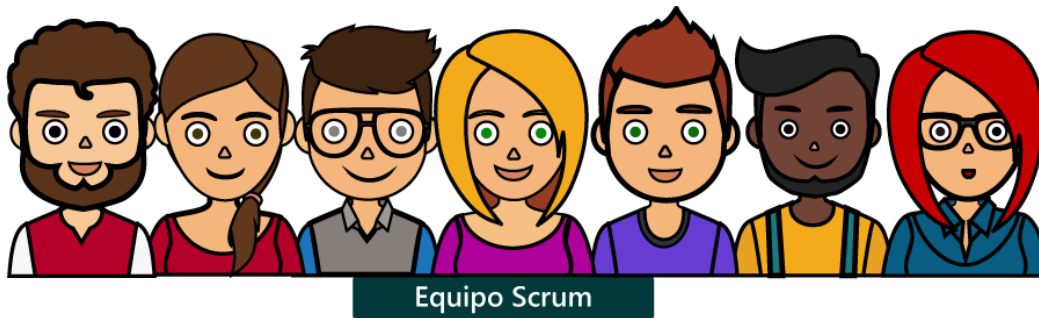


Ilustración 3 - El Equipo Scrum

Un Equipo Scrum consiste en un Propietario del Producto (Product Owner), el Equipo de Desarrollo y un Scrum Master.

Los Equipos Scrum son autoorganizados y multifuncionales. El modelo de Equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. El Equipo Scrum ha demostrado ser incrementalmente efectivo en diferentes contextos y para cualquier trabajo complejo.

Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades para poder obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

2.1 - El Product Owner



Ilustración 4 - Product Owner

El Product Owner es el responsable de maximizar el valor entregado por el Equipo de Desarrollo.

El Product Owner es la única persona responsable de gestionar el Product Backlog. La gestión del Product Backlog incluye:

- Expresar claramente los elementos del Product Backlog.
- Ordenar los elementos en el Product Backlog para alcanzar los objetivos y las misiones de la mejor manera posible.
- Garantizar que el Product Backlog sea visible, transparente y clara para todos y que muestre, lo que el equipo trabajará a continuación.
- Asegurar que el Equipo de Desarrollo entiende los elementos del Product Backlog a nivel necesario.
- El Product Owner es una única persona, no un comité. El Product Owner podría representar los deseos de un comité en el Product Backlog, pero aquellos que quieran cambiar la prioridad de un elemento del Product Backlog deben hacerlo a través del Product Owner.

Para que el Product Owner pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Product Owner se reflejan en el contenido y en la priorización del Product Backlog.

El Product Owner también es responsable de:

- Participar en las reuniones de apertura y cierre del proyecto.
- Gestionar los riesgos globales del proyecto.
- Presentar informes del proyecto al cliente u otras partes interesadas.
- Aprobar cambios en el proyecto.

- Asegurar que se administran correctamente los recursos financieros del proyecto al inicio y durante su ejecución.

Normalmente al Product Owner se le asocia con un Gerente de Proyecto, dadas sus responsabilidades de gestión, aunque es más preciso asociarlo al rol que representa “La voz del cliente”.

2.2 - El Equipo de Desarrollo



Ilustración 5 - El Equipo de Desarrollo

El Equipo de Desarrollo se compone de profesionales que realizan el trabajo de entregar un Incremento de producto “Terminado” (Done) que potencialmente se pueda poner en producción al final de cada Sprint. Un Incremento de producto “Terminado” es obligatorio en la Revisión del Sprint (*ver sección 3*). Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

La organización es la encargada de estructurar y empoderar a los Equipos de Desarrollo para que estos organicen y gestionen su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

- Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos del Product Backlog en Incrementos de funcionalidad potencialmente desplegables.
- Nadie fuera del Scrum Master y el Product Owner puede pedir al Equipo de Desarrollo que trabaje en un conjunto diferente de requisitos.
- Los Equipos de Desarrollo son multifuncionales, con todas las habilidades necesarias para crear un Incremento de producto.
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, independientemente del trabajo que realice cada persona.

- Scrum no reconoce sub-equipos en los Equipos de Desarrollo, no importan los dominios particulares que requieran tenerse en cuenta, como pruebas, arquitectura, operaciones, o análisis de negocio.
- Los miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

2.2.1 - Tamaño del Equipo de Desarrollo

El tamaño óptimo del Equipo de Desarrollo es de entre **3 y 9** miembros, esto le permite ser lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para poder completar una cantidad significativa de trabajo.

- Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas.
- Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción.
- Tener más de nueve miembros en el equipo requiere demasiada coordinación.
- Los Equipos de Desarrollo grandes generan demasiada complejidad como para que un proceso empírico pueda ser de utilidad.
- Los roles de Product Owner y Scrum Master no se contabilizan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en el Sprint Backlog.

2.3 El Scrum Master (Scrum Master)



Ilustración 6 - El Scrum Master

El Scrum Master es el responsable en promocionar y apoyar Scrum como se define en la Guía de Scrum. Su principal responsabilidad entonces es garantizar que todos conocen y aplican correctamente la teoría de Scrum, sus prácticas y sus reglas.

- El Scrum Master es un líder sirviente que está al servicio del Equipo Scrum.
- El Scrum Master ayuda a las personas externas al Equipo Scrum (*involucrados 6.1.1.2*) a entender qué interacciones con el Equipo Scrum pueden ser útiles y cuáles no.
- El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

2.3.1 - El servicio del Scrum Master al Propietario del Producto (Product Owner)

El Scrum Master sirve al Product Owner de varias formas, incluyendo:

- Asegurar que los objetivos, el alcance y el dominio del producto sean entendidos por todos en el Equipo de la mejor manera posible.
- Encontrar técnicas para gestionar el Product Backlog de manera efectiva.
- Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Product Backlog claros y concisos.
- Entender la planificación del producto en un entorno empírico.
- Asegurar que el Product Owner conozca cómo ordenar el Product Backlog para maximizar el valor.
- Entender y practicar la agilidad.
- Facilitar los eventos de Scrum según se requiera o necesite.

2.3.2 - El servicio del Scrum Master al Equipo de Desarrollo

El Scrum Master sirve al Equipo de Desarrollo de varias formas, incluyendo:

- Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional.
- Ayudar al Equipo de Desarrollo a crear productos de alto valor.
- Eliminar impedimentos para el progreso del Equipo de Desarrollo.
- Facilitar los eventos de Scrum según se requiera o necesite.
- Guiar al Equipo de Desarrollo en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

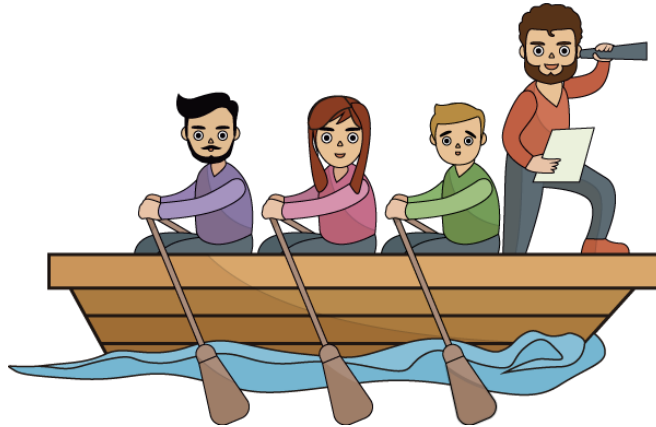


Ilustración 7 - El Scrum Master guía al equipo

2.3.3 - El servicio del Scrum Master a la Organización

El Scrum Master sirve a la organización de varias formas, incluyendo:

- Liderar y guiar a la organización en la adopción de Scrum.
- Planificar las implementaciones de Scrum en la organización.
- Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto.
- Motivar cambios que incrementen la productividad del Equipo Scrum.
- Junto con otros Scrum Masters, incrementar la efectividad de la aplicación de Scrum en la organización.

2.4 - Los valores del equipo

Es importante que entre los miembros del equipo se genere y mantenga una filosofía basada en valores que fomenten la confianza, la comunicación y la entrega de resultados. A continuación, se relacionan los valores que deberían existir en un equipo Scrum.

2.4.1 – Compromiso

El compromiso se refiere a que cada miembro del Equipo hará el máximo esfuerzo posible, además pondrá dedicación a todas las actividades del proyecto.

2.4.2 – Respeto

Los miembros del Equipo Scrum respetan el conocimiento, las habilidades y la experiencia profesional no solo del resto de miembros del equipo, sino también de aquellas personas con las que se relacionan, sean de su propia organización o de otra.

2.4.3 - Proactividad

La proactividad les permite a los miembros del Equipo el desarrollo exitoso de su trabajo. Este valor es fundamental para lograr buenos resultados en los entornos cambiantes a los que se encuentran expuestos los proyectos ágiles, este valor no se limita a la adaptabilidad (responder al cambio), la proactividad se trata de iniciar el cambio.

2.4.4 – Receptividad - Apertura

Los miembros del Equipo Scrum deben estar abiertos a aprender nuevas habilidades o adquirir nuevos conocimientos que les permitan ser Equipos multi-funcionales. Este valor también está relacionado con la transparencia, la colaboración y el libre conocimiento.

2.4.5 – Foco

Este principio les permitirá a los miembros del Equipo Scrum centrarse en solo aquellas actividades clave que permitirán ofrecer el mejor producto o servicio, el foco se trata de centrarse en todo aquello que es realmente importante no sólo para el proyecto sino también para el bienestar del equipo.

3. Eventos en Scrum

En Scrum existen diferentes eventos predefinidos con el fin de generar regularidad y minimizar la necesidad de reuniones no definidas en Scrum.

Todos los eventos son compartimientos o periodos de tiempo limitado (time-boxes), de tal modo que todos tienen una duración máxima.

3.1 - El Sprint (Sprint)

El corazón de Scrum es el Sprint, es un compartimiento o periodo de tiempo (time-box) de **1 a 4 semanas** durante el cual se crea un incremento de producto "Terminado" utilizable y potencialmente desplegable.

Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint anterior. Los Sprints contienen y consisten en la Planificación del Sprint, los Scrums Diarios, el trabajo de desarrollo, la Revisión del Sprint, y la Retrospectiva del Sprint.

Durante el Sprint:

- No se realizan cambios que puedan afectar al objetivo del Sprint.
- Los objetivos de calidad no disminuyen.
- El alcance puede clarificarse y renegociarse entre el Product Owner y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para alcanzar algo. Cada Sprint tiene un objetivo de lo que se construirá, un diseño y un plan flexible que guiará su construcción, el trabajo del equipo y el incremento de producto resultante.

Los Sprints están limitados a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande, la definición de lo que se está construyendo podría cambiar, la complejidad podría incrementarse y el riesgo podría aumentar.

Los Sprints habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario.

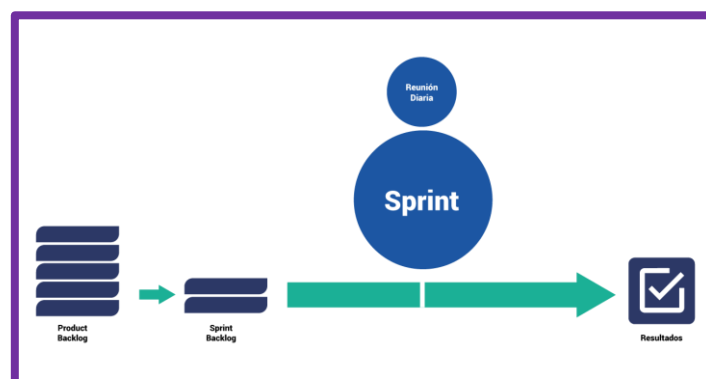


Ilustración 8 - El Sprint

3.1.1 - Consideraciones sobre la duración del Sprint

- Una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse.
- Los otros eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.
- Todos los eventos relacionados con el Sprint deben ejecutarse, la falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida de inspección y adaptación.
- Dentro de un mismo proyecto, los Sprints pueden tener duración variable (conservando la regla de 1 a 4 semanas). La duración de los Sprint dependerá de los siguientes factores:
 - Fechas de entrega acordadas
 - Experiencia del equipo (a menor experiencia con Scrum, los Sprints deberían ser más cortos)
 - Restricciones propias del proyecto
- La duración del Sprint es definida por el Equipo de Desarrollo con las recomendaciones del Scrum Master, quienes a su vez deberán llegar a un consenso con el Product Owner, por lo general esto se hace durante la Definición del Cronograma de Entregas (7.1.5).

3.1.2 - Cancelación del Sprint

Un Sprint puede cancelarse antes que el periodo de tiempo llegue a su fin. Algunas de las razones por las que se podría cancelar un Sprint son:

- La organización cambia la dirección del proyecto y eso afecta el entregable del Sprint.
- Las condiciones del mercado o de la tecnología cambian.
- El objetivo del Sprint quedó obsoleto.
- Surge un error sobre un producto que está en ambiente de producción y el equipo de desarrollo debe resolverlo cuanto antes.
- Surge un cambio urgente que debe introducirse de inmediato.

Algunas consideraciones que se deben tener en cuenta sobre la cancelación de los Sprint son:

- Solo el Product Owner tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de los interesados (7.1.1.2), del Equipo de Desarrollo o del Scrum Master.

- Debido a la corta duración de los Sprints, su cancelación rara vez tiene sentido.
- Cuando se cancela un Sprint se revisan todos los Elementos del Product Backlog que se hayan completado y "Terminado". Si una parte del trabajo es potencialmente entregable, el Product Owner normalmente la acepta.
- Todos los elementos del Product Backlog no completados se vuelven a estimar y se vuelven a introducir en el Product Backlog. El trabajo finalizado en ellos pierde valor con rapidez y por lo general debe volverse a estimar.
- Las cancelaciones de Sprint consumen recursos ya que todos se reagrupan en otra Planificación de Sprint para empezar otro Sprint.
- Las cancelaciones del Sprint son a menudo traumáticas para el Equipo Scrum y son muy poco comunes.

3.2 – Reunión de Planificación del Sprint

- En este evento se planifica el trabajo a realizar durante un Sprint.
- Este plan se crea mediante el trabajo colaborativo de todo el Equipo Scrum.
- El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito.
- El Scrum Master enseña al Equipo Scrum a mantenerse dentro del periodo de tiempo.

3.2.1 - Datos del evento

- **Duración:** 2 horas por cada semana de Sprint a planificar. (Duración máxima de ocho horas).
- **Participantes:** Todo el Equipo Scrum (*Sección 2*).
- **¿Qué se hace en la reunión?**
 - o 1. ¿Qué puede terminarse en este Sprint?
 - 1.1 – Se identifican los cambios que deben introducirse en este Sprint (*Sección 6.4*).
 - 1.2 - Seleccionar los elementos del Product Backlog.
 - 1.3 – Identificar las tareas que deben ejecutarse para "terminar" esos elementos.
 - 1.4 – Identificar las dependencias que puedan existir.
 - 1.5 – Se realiza la estimación de los elementos seleccionados para su desarrollo.
 - 1.6 – El equipo de desarrollo se autoasigna los elementos del Product Backlog.
 - o 2. ¿Cómo se conseguirá completar el trabajo seleccionado?
 - 2.1 - ¿Qué riesgos podrán afectarnos este Sprint? (*Sección 6.3*)

- 3. Formalizar el objetivo del Sprint
 - El objetivo del Sprint se plasma en el Sprint Backlog (*Sección 4.2 - Sprint Backlog*)

3.3 - Daily Scrum (Scrum Diario)

El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para el Equipo de Desarrollo. El Scrum Diario se realiza diariamente para cada día del sprint.

3.3.1 - Datos del evento

- **Duración:** 15 minutos máximo.
- **Participantes:** El Equipo de Desarrollo y el Scrum Master
- **¿Qué se hace en la reunión?**

El Equipo de Desarrollo responde las siguientes preguntas:

- ¿Qué hice hoy?
 - ¿Qué haré mañana?
 - ¿Existe algún impedimento que está afectando o nos pueda afectar?
-
- Durante el Scrum Diario el Equipo de Desarrollo planea el trabajo para las siguientes 24 horas.
 - El Scrum Diario optimiza la colaboración y el desempeño del equipo inspeccionando el trabajo avanzado desde el último Scrum Diario y proyectando el trabajo a realizar a continuación.
 - El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum Diario, para tener discusiones detalladas, o para adaptar o replanificar el resto del trabajo del Sprint.
 - El Scrum Diario se realiza a la misma hora y lugar todos los días para reducir la complejidad. Por lo general se realiza antes de finalizar la jornada de trabajo del Equipo de Desarrollo.
 - El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint.
 - El Scrum Diario optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint.
 - El Scrum Master se asegura de que el Equipo de Desarrollo tenga la reunión, pero es el Equipo de Desarrollo el responsable de dirigir el Scrum Diario, es decir que, no es obligatoria la presencia del Scrum Master para que se lleve a cabo el Scrum Diario.

- El Scrum Master enseña al Equipo de Desarrollo a mantener el Scrum Diario en los límites del bloque de tiempo de 15 minutos.
- El Scrum Diario es una reunión interna del Equipo de Desarrollo. Si otras personas están presentes, el Scrum Master se asegura de que no interrumpen la reunión.
- Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de realizar otras reuniones, identifican impedimentos a remover relativos al desarrollo, resaltan y promueven la toma rápida de decisiones y mejoran el nivel de conocimiento del Equipo de Desarrollo.
- Durante los Scrum Diarios se actualizan el Burndown chart y el Scrum Board.
- Esta reunión por lo general se realiza de pie con el fin de garantizar el cumplimiento del tiempo asignado, por lo que también se le puede conocer como el Scrum Diario de Pie (Daily Standup Meeting).

3.4 – Reunión de Revisión del Sprint

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar el Product Backlog si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto y en cualquier cambio al Product Backlog durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado.

3.4.1 Datos del evento

- **Duración:** 1 hora por cada semana de Sprint a revisar. (Duración máxima de cuatro horas).
- **Participantes:** Todo el Equipo Scrum (*Sección 2*) + Los Interesados clave que sean invitados por el Product Owner
- **¿Qué se hace en la reunión?**
 - El Equipo de Desarrollo habla acerca de qué estuvo bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas.
 - El Equipo de Desarrollo hace una demostración del trabajo que ha “Terminado” y responde preguntas acerca del Incremento y su funcionalidad.

- El Product Owner habla acerca del Product Backlog en su estado actual. Proyecta objetivos probables y fechas de entrega en el tiempo basándose en el progreso obtenido hasta la fecha (si fuera necesario).
- El Product Owner realiza la aprobación de los elementos del Product Backlog que se han "Terminado" y rechaza los que no se han "Terminado" y con esto determinar la Deuda Técnica.
- El equipo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.
- Revisión de la cronología, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

El resultado de la Revisión de Sprint es un Product Backlog revisado que define los elementos del Product Backlog posibles para el siguiente Sprint. Es posible además que el Product Backlog reciba un ajuste general para enfocarse en nuevas oportunidades.

3.5 – Reunión de Retrospectiva del Sprint

El tema principal de la Reunión de Retrospectiva del Sprint es la identificación de posibles mejoras que puede incorporar el Equipo en Próximos Sprints.

3.5.1 - Datos del evento

- **Duración:** 1 hora por cada semana de Sprint a revisar. (Duración máxima de cuatro horas).
- **Participantes:** Equipo de Desarrollo y el Scrum Master (no se recomienda que el Product Owner haga parte de esta reunión).
- **¿Qué se hace en la reunión?**
 - Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas.
 - Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras que puede incorporar el Equipo en Próximos Sprints.
 - Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

4. Los Artefactos en Scrum

Los artefactos de Scrum representan el trabajo o el valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

4.1 - Product Backlog

El Product Backlog es una lista ordenada de todo lo conocido que podría ser necesario en el producto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Product Owner es el responsable del Product Backlog, incluyendo su contenido, disponibilidad y ordenación.

Un Product Backlog nunca está completo. El desarrollo más temprano del mismo solo refleja los requisitos conocidos y mejor entendidos al principio.

El Product Backlog evoluciona a medida que el producto y el entorno en el que se usará también lo hacen. El Product Backlog es dinámico; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Product Backlog también existe.

El Product Backlog enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a realizarse sobre el producto para entregas futuras. Los elementos del Product Backlog tienen como mínimo los siguientes atributos:

- La descripción (en función del valor para el negocio)
- El orden (prioridad)
- La estimación
- Los criterios de aceptación

A medida que un producto es utilizado y se incrementa su valor y el mercado proporciona retroalimentación, el Product Backlog se convierte en una lista más larga y exhaustiva. Los requisitos nunca dejan de cambiar así que el Product Backlog es un artefacto vivo. Los cambios en los requisitos de negocio, las condiciones del mercado o la tecnología podrían causar cambios en el Product Backlog.

¡Nota Importante! Para evitar proyectos que nunca terminan, es importante considerar el alcance del proyecto a la hora de refinar el Product Backlog.

4.2 - Sprint Backlog

El Sprint Backlog es el conjunto de los elementos del Product Backlog seleccionados para un Sprint, más un plan para entregar el Incremento de producto y conseguir el objetivo del Sprint.

El Sprint Backlog es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento "Terminado".

Cuando se requiere nuevo trabajo, el Equipo de Desarrollo lo adiciona al Sprint Backlog. A medida que el trabajo se ejecuta o se completa se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan se considera innecesario, es eliminado. Solo el Equipo de Desarrollo puede cambiar su Sprint Backlog durante un Sprint. El Sprint Backlog es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante el Sprint y pertenece únicamente al Equipo de Desarrollo.

- El Sprint Backlog hace visible todo el trabajo que el Equipo de Desarrollo debe completar para alcanzar el Objetivo del Sprint.
- Para asegurar el mejoramiento continuo, en el Sprint Backlog se incluye por lo menos una actividad que permita la mejora de procesos (normalmente identificadas en la Retrospectiva inmediatamente anterior).
- El Sprint Backlog es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Scrum Diario.

4.2.1 - Incremento

Un Incremento es la suma de todos los elementos del Product Backlog completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint el nuevo Incremento debe estar "Terminado", lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de "Terminado" del Equipo Scrum.

- El incremento es un paso hacia la visión o meta del proyecto.

- El incremento debe estar en condiciones de utilizarse sin importar si el Product Owner, decide liberarlo o no.

4.3 - Emisores de Información

Los emisores de información son artefactos que permiten visualizar gráficamente el estado de los Sprints o del proyecto en general. Sirven para garantizar la transparencia y para realizar seguimiento y control del proyecto (Sección 6.2).

4.3.1 - Scrum Board

El Scrum Board o también conocido como tablero Scrum es una adaptación del tablero Kanban que nos permite dar seguimiento a cada Sprint Backlog dentro de un proyecto Scrum.

El Scrum Board es un emisor de información que le permite al Equipo garantizar la transparencia en los Sprints, mantiene la coordinación y permite realizar al Scrum Master sus labores de Inspección. (Sección 5.1)

Está compuesto por 4 columnas: Pendiente, En Progreso, En Prueba (En Revisión), "Terminado".

Cada elemento por desarrollar debe ponerse en una tarjeta individual, y dado que el Scrum Board se actualiza constantemente, todas las tarjetas deberán pasar por las 4 columnas. (No se pueden hacer saltos de columnas).

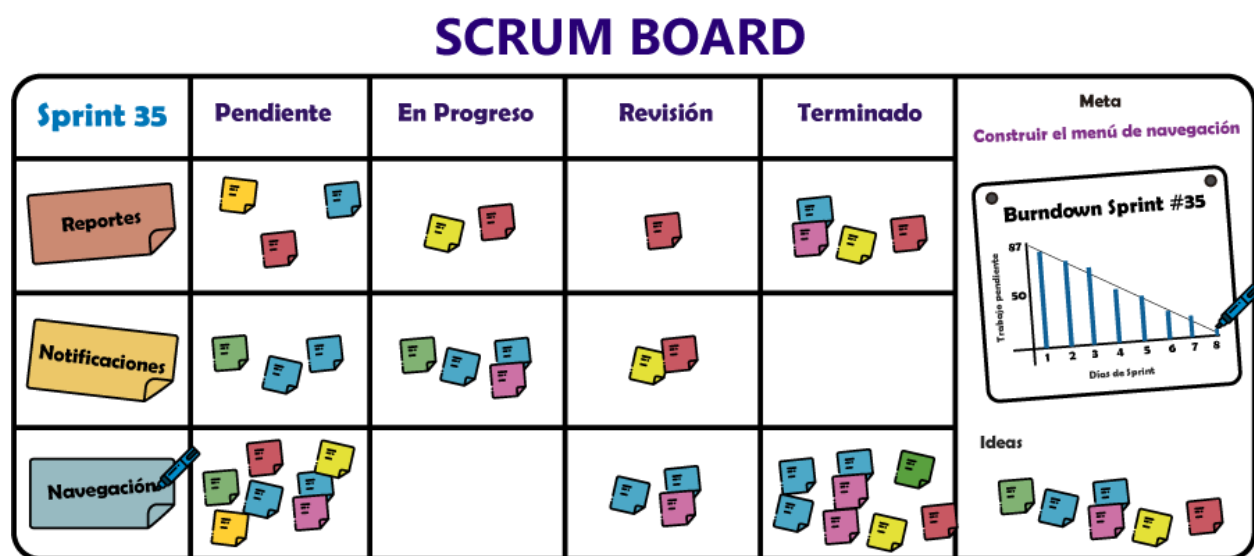


Ilustración 9 - Scrum Board

Aclaraciones sobre el Scrum Board

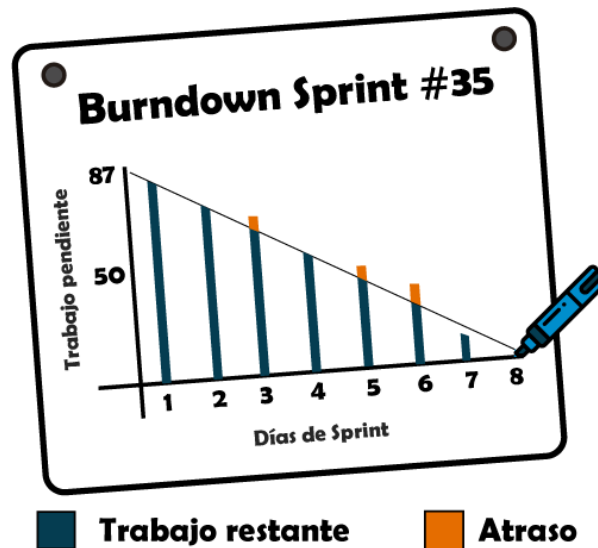
- Cuando el tablero está físico en el lugar de trabajo del equipo, generalmente las tarjetas que se manejan son “sticky notes” de distintos colores.
- Cuando se van a desarrollar múltiples historias de usuario en un solo Sprint conviene dividir el tablero en filas, donde cada fila representa un componente o épica del producto.
- Al final de un Sprint se “borra” el Scrum Board
- Para considerar “terminado” un elemento y moverlo a esta zona en el Scrum Board, se debe considerar la “definición de terminado (DoD)” (Sección 7.1.1.3).
- Para facilitar la lectura del Scrum Board, a cada miembro del equipo puede asignársele un color de tarjetas.
- Aunque por lo general el Scrum Board se actualiza durante la reunión diaria, cada integrante del Equipo de Desarrollo tiene autonomía para actualizar su conjunto de elementos asignados en el Sprint Backlog.

4.3.2 - Burndown chart del Sprint

El “Burndown Chart” es un emisor de información que muestra la cantidad de trabajo pendiente que queda en el Sprint actual.

Es un emisor gráfico de 2 dimensiones:

- El eje vertical se construye a partir de la sumatoria de puntos de historia a desarrollar en el Sprint
- El eje horizontal corresponde a la duración del Sprint en días hábiles.

*Ilustración 10 - Burndown chart del Sprint*

Aclaraciones sobre el Burndown Chart

- Una posible variación es el Burnup chart que muestra el trabajo completado en el Sprint
- El Equipo de Desarrollo es el responsable de la actualización del Burndown Chart
- Por lo general la actualización se realiza durante los Scrum Diarios.

4.3.3 – Diagrama de flujo acumulado

El Diagrama de Flujo Acumulado (CFD - Cumulative Flow Diagram) es un emisor de información bastante útil para la elaboración de informes y el seguimiento de los resultados del proyecto.

Este emisor de información muestra el progreso del proyecto respecto a los ítems del Product Backlog.

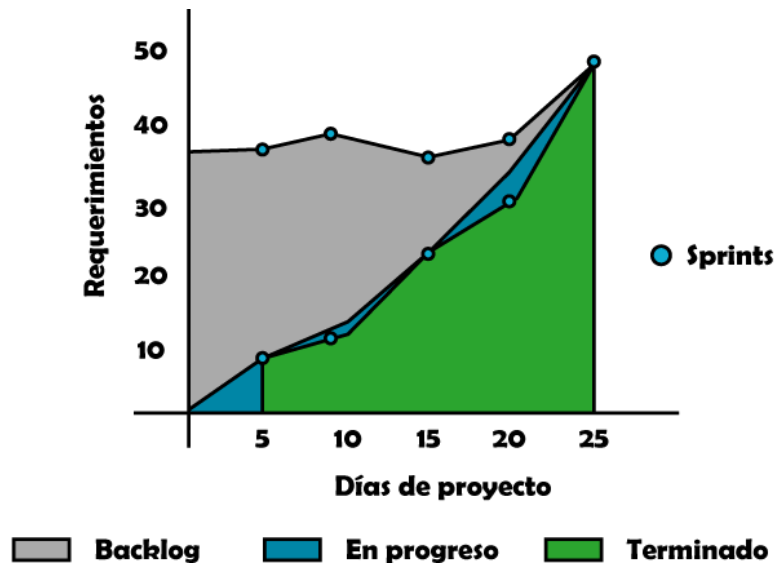


Ilustración 11 - Diagrama de flujo acumulado

- La zona "gris" muestra el comportamiento del Product Backlog.
 - Los incrementos significan la adición de nuevos requerimientos/tareas/historias de usuario.
 - Los decrementos significan el retiro de requerimientos/tareas/historias de usuario que ya no generan valor (producto de cambios).
- La zona "azul" muestra el trabajo que fue seleccionado por el equipo para desarrollar en los distintos Sprints.
- La zona "verde" muestra el trabajo terminado en el proyecto alrededor del tiempo.
 - Cuando la zona azul está por encima de la zona verde, normalmente significa que el equipo seleccionó más trabajo del que podía terminar, esto se conoce como "deuda técnica".
 - La diferencia entre la zona "verde" y la zona "gris", está relacionada con el progreso del proyecto.
- Los puntos marcados en el gráfico muestran los distintos Sprints.
- El eje vertical podría ser el "tamaño de los requerimientos", esto permitiría tener mayor precisión a la hora de calcular el progreso del proyecto.
- El diagrama de flujo acumulado es único por proyecto.
- La técnica originalmente es de la metodología Kanban.
- El responsable de este radiador de información es el Scrum Master, aunque normalmente es el Product Owner quien lo utiliza para mostrar el progreso del proyecto a las partes interesadas.

4.4 – Registro de obstáculos/impedimentos

El registro de obstáculos, o también llamado registro de impedimentos es un artefacto en el que se registran todos los obstáculos que se presentan en los proyectos y su respectiva solución.

Este artefacto es responsabilidad de los Scrum Masters y se actualiza por lo general durante los Scrum Diarios.

En algunas organizaciones este artefacto es global para todos los proyectos y sirve como base de conocimiento para todos los integrantes de los distintos equipos Scrum.

5. Principios de Scrum

Scrum define 6 principios que son clave para el buen funcionamiento del Marco de Trabajo. Se puede asegurar que son las “reglas” que deben conocer y aplicar los miembros del Equipo Scrum para garantizar el correcto funcionamiento de este marco de trabajo.

5.1 - Control empírico de procesos (1)

Scrum se basa en la teoría de control empírico de procesos o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo.

Los 3 pilares que soportan toda la implementación del control empírico de procesos son: transparencia, inspección y adaptación.

5.1.1 - Transparencia

Los aspectos significativos del proyecto deben ser visibles para las partes interesadas. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.

Por ejemplo:

- Todos los participantes deben compartir un lenguaje común para referirse al proceso.

- Aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de “Terminado”.

5.1.1.1 - Transparencia de los Artefactos

Scrum se basa en la transparencia. Las decisiones para optimizar el valor y controlar el riesgo se toman basadas en el estado percibido de los artefactos. En la medida en que la transparencia sea completa, estas decisiones tienen unas bases sólidas. En la medida en que los artefactos no son completamente transparentes, estas decisiones pueden ser erróneas, el valor puede disminuir y el riesgo puede aumentar.

El Scrum Master debe trabajar con el Product Owner, el Equipo de Desarrollo y otras partes interesadas para entender si los artefactos son completamente transparentes. Hay prácticas para hacer frente a la falta de transparencia; el Scrum Master debe ayudar a todos a aplicar las prácticas más apropiadas si no hay una transparencia completa.

Un Scrum Master puede detectar la falta de transparencia inspeccionando los artefactos, reconociendo patrones, escuchando atentamente lo que se dice y detectando diferencias entre los resultados esperados y los reales.

La labor del Scrum Master es trabajar con el Equipo Scrum y la organización para mejorar la transparencia de los artefactos. Este trabajo usualmente incluye aprendizaje, convicción y cambio. La transparencia no ocurre de la noche a la mañana, sino que es un camino.

5.1.2 - Inspección

En los proyectos Scrum se deben inspeccionar frecuentemente los artefactos y el progreso hacia un objetivo, para detectar variaciones.

La inspección no debe ser tan frecuente como para que interfiera en el trabajo, las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos en el mismo lugar de trabajo (normalmente el Scrum Master).

5.1.3 - Adaptación

Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación, tal y como se describen en la sección Eventos de Scrum de esta guía:

- Reunión de Planificación del Sprint.
- Scrum Diario.
- Revisión del Sprint.
- Retrospectiva del Sprint.

5.2– Autoorganización (2)

Para Scrum, El Equipo de Desarrollo debe ser autoorganizado, esto significa que son los miembros del equipo quienes eligen la mejor opción de llevar a cabo su trabajo sin ser dirigidos por personas externas al equipo.

A continuación, se listan las reglas que garantizan la autoorganización del Equipo:

- El Equipo de Desarrollo es quien se autoasigna el trabajo a realizar en los diferentes Sprints, nadie, ni siquiera el Product Owner debe imponer el trabajo al Equipo.
- El Equipo debe conocer muy bien sus límites de decisión para así poder tener mayor autonomía.
- El Equipo debe tener espacios que le permitan realizar jornadas de investigación y capacitación.
- El Equipo debe mantenerse motivado.

Para que un equipo pueda ser autoorganizado, se necesitan como mínimo los siguientes elementos:

- Debe existir una meta formalmente definida y de conocimiento para el equipo, "si el equipo no tiene un rumbo definido, no sabrá a donde debe llegar y le será imposible autoorganizarse".
- El Equipo debe entender la visión del proyecto y por qué el proyecto aporta valor a la organización.
- La transparencia y la inspección son fundamentales para un equipo autoorganizado.
- Los integrantes del Equipo deben ser multifuncionales, además de estar actualizando sus conocimientos y habilidades de manera continua.

5.2.1 - Colaboración de equipo

La colaboración se da gracias a la constante comunicación que existe en los Equipos Scrum, tanto entre sus miembros como con las Partes interesadas del Proyecto, este concepto es parte integral del Manifiesto Ágil *"La forma más eficiente y efectiva de transmitir información hacia y dentro del Equipo de Desarrollo es la conversación cara a cara"*.

El Scrum Master es el rol responsable de garantizar una sana comunicación entre todas las partes interesadas del proyecto, en especial de su Equipo de Desarrollo.

Se debe considerar que, según la naturaleza del proyecto, las necesidades de la organización e incluso factores externos, determinan la ubicación de los miembros del Equipo. Es por esto que en Scrum los Equipos se clasifican en 2 categorías:

5.2.1.1 - Equipos Centralizados

Las características de un Equipo Centralizado son:

- Los miembros del Equipo se encuentran en la misma ubicación, lo que les permite comunicarse con gran facilidad.
- La resolución de problemas es prácticamente inmediata, ya que al estar ubicados en el mismo lugar es fácil realizar sesiones de diálogo.

5.2.1.2 - Equipos Distribuidos

Un Equipo Distribuido con se encuentra en una misma ubicación, por lo general está disperso debido a la subcontratación, diferentes ubicaciones físicas, las opciones de trabajo desde casa, etc.

Para garantizar la comunicación permanente en este tipo de equipos se hacen necesarias las siguientes herramientas:

- Groupware.
- Software Videollamadas o chat.
- Software de gestión de proyectos ágiles.
- Herramientas de software que simulan la funcionalidad de Scrum boards.

5.2.2 - Colaboración con el cliente

En los proyectos tradicionales, los clientes por lo general se mantenían a distancia y solo se involucraban al principio y al final del proyecto. En Scrum es altamente recomendable que el cliente participe de las revisiones del producto y brinde retroalimentación en todos los puntos de "inspección y adaptación". Esto minimiza el riesgo y le brinda más opciones al cliente y a las partes interesadas.

Por ejemplo, en otros Marcos Ágiles como XP, es obligatorio que el Cliente forme parte del equipo.

El cliente (o sus representantes) deberían trabajar junto al Product Owner para definir las historias de usuario y detallar dichas historias antes o durante las reuniones de planificación.

El cliente y las partes interesadas por lo general participan en la Reunión de Revisión de los Sprint y, dependiendo de la relación entre el cliente y el Product Owner, el Cliente incluso podría participar de algunas reuniones de Retrospectiva de los Sprint.

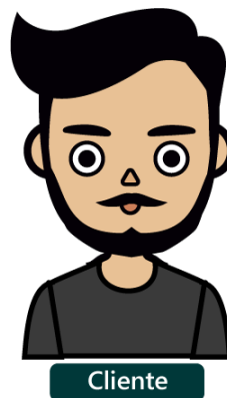


Ilustración 12 - El cliente participa en el proyecto ágil

5.2.3 - Equipos multifuncionales

Los equipos multifuncionales tienen todas las competencias y habilidades necesarias para llevar a cabo el trabajo sin depender de otras personas que no formen parte del equipo.

Contrario a lo que piensa un equipo multifuncional, no se trata de que todos sus integrantes hagan de todo, se trata de que los integrantes adquieran conocimiento en distintas disciplinas (aplicables a los proyectos de la organización) y así puedan contribuir eficazmente con la colaboración.

La realidad es que incluso cuando un equipo sea experto técnico, siempre necesitarán capacitación adicional, así es que el Product Owner deberá decidir si aprobará el dinero y el tiempo para capacitarse o por el contrario serán los miembros del equipo quienes se encargarán del tema (*Sección 6.1*).

5.2.3.1 – Gestionar el conocimiento

Realizar este ejercicio permitirá identificar, recopilar, organizar, transferir y retener el conocimiento necesario para dar soporte a todo el personal en sus actividades laborales, para la toma de decisiones bien fundadas y para aumentar la productividad.

5.2.4 - Propiedad colectiva del producto

Para garantizar una colaboración constante y evitar con el tiempo la aparición de una cultura de la culpa, es importante fomentar una propiedad colectiva del producto, esto significa que todo el Equipo Scrum es dueño del producto, y por tanto cualquiera de sus integrantes podría contribuir al desarrollo de cualquier parte del producto aun cuando no haya sido quien lo desarrolló inicialmente.

Así mismo no deberían existir reconocimientos individuales a los miembros del equipo por sus contribuciones al producto.

5.2.5 – Motivación del equipo

Los equipos Scrum se caracterizan por mantener un enfoque hacia la entrega frecuente de resultados; y aunque los miembros del equipo son conscientes de la responsabilidad que esto implica, existe un factor de fondo que facilita el impulso y el esfuerzo para cumplir con los objetivos; la motivación.

La motivación hace referencia a que los miembros del equipo mantengan determinada conducta y estado de ánimo que propicien las interacciones sanas y el alto rendimiento en el proyecto.

Tipos de motivación

- **Intrínseca:** este tipo de motivación es propio de cada persona, es decir que por su propia voluntad e inspiración es capaz de mantener una conducta específica y el impulso necesario para cumplir con una meta que brinda satisfacción interna y realización personal.

- **Extrínseca:** este tipo de motivación hace referencia a mantener una conducta específica para responder a un impulso externo, es decir que en este caso la voluntad e inspiración de la persona se ven influenciadas por una recompensa externa (que puede ser algo físico, monetario o psicológico).
Según cómo se maneje la motivación en el equipo Scrum, eventualmente la motivación extrínseca tiende a convertirse en motivación intrínseca, pues los miembros del equipo van adaptando su conducta y mejorando su rendimiento para cumplir los objetivos sin necesidad de que todo el tiempo estén recibiendo recompensas o algo a cambio.

Identificar los motivadores

Cuando hablamos de motivación, es el Scrum Master quien tendrá la mayor participación y responsabilidad pues al estar al servicio del equipo Scrum puede identificar fácilmente qué es lo que aumenta o disminuye la motivación de los miembros del equipo; es por esto que dentro de las tareas del Scrum Master se incluye la identificación y análisis de los elementos motivadores del equipo para construir y desarrollar un Plan de Motivación que posteriormente negociará con el Product Owner para asegurar los recursos necesarios para su ejecución.

Incentivar la investigación

Dentro del conjunto de factores que aumentan la motivación del equipo en Scrum se recomienda impulsar la investigación de nuevos productos o tecnologías lo cual favorece que los miembros del equipo adquieran nuevo conocimiento y se inclinen hacia el logro de nuevos objetivos.

La investigación también propiciará que los miembros del equipo se propongan metas basadas en la autorrealización y desarrollo de sus propias competencias.

Algunas metodologías en el mercado hablan de las técnicas que pueden incentivar la investigación en los equipos, tales como: DemoDay – Hackathon – Exploration Days – TED days, etc.

5.3 – Simplicidad (3)

Scrum no sería considerada una metodología ágil de no ser por su simplicidad, es por ello que se intenta al máximo reducir la burocracia en sus prácticas, se trabaja con los artefactos que son esenciales para el

proyecto y se sigue un flujo de prácticas simple, sin descuidar todos los elementos necesarios para la correcta gestión del proyecto.

- Es responsabilidad del Scrum Master garantizar que la simplicidad será el pilar fundamental para la adopción de Scrum.

5.3.1 - Diseño simple

Lo más importante en el desarrollo de un producto usando el marco Scrum, es el valor que entrega para los clientes/usuarios, considerando además que su desarrollo debería tardar el menor tiempo posible.

Es por esto que es altamente recomendable establecer un alcance para el proyecto que considere suficientes características para que el producto sea de alto valor para el usuario pero que tenga el menor tiempo de desarrollo posible, sin descuidar la calidad.

Este concepto tiene su justificación en los principios del Manifiesto Ágil *“El producto funcional es más importante que la documentación exhaustiva”*. (Ver más en la sección 6.5)

5.3.2 - Uso de herramientas de software

Las herramientas de software para la gestión de proyectos ágiles se han convertido en un elemento indispensable para garantizar la simplicidad. Algunas de las ventajas de usar estas herramientas son:

- Centralizan la información de los proyectos, permitiendo un mejor control de la información.
- Permiten automatizar tareas, por ejemplo:
 - Estimaciones basadas en históricos.
 - Calcular la velocidad del equipo.
 - Elaborar gráficos para el seguimiento del presupuesto, progreso del Sprint (Burndown), progreso del proyecto (Diagrama de flujo acumulado), etc.
 - Generar actas de reuniones.
- Generan notificaciones sobre elementos del Product Backlog con atraso.
- Facilitan la interacción de los Equipos Distribuidos geográficamente.
- Si los proyectos son de desarrollo de software, permiten, además:
 - Integración continua.
 - Pruebas automatizadas.

- Trazabilidad bidireccional entre código e historias de usuario.

5.3.3 - ¿Vale la pena trabajar con procesos?

Si bien, en muchas organizaciones trabajar con procesos es visto como una gran ventaja porque asegura el buen desempeño de sus colaboradores, cuando hablamos de proyectos ágiles, tendremos que ver la situación desde otro punto.

Las organizaciones cuentan con modelos de operación que tienden a comportarse de manera “estática”, en los que generalmente encontramos una estructura de jerarquías y burocracia, donde de manera rutinaria los colaboradores conocen los productos/servicios de la organización y saben cómo operar en el día a día para garantizar su adecuada entrega; sin embargo con los proyectos no siempre ocurre lo mismo debido a que los miembros del equipo de proyecto comúnmente se enfrentan a nuevos desafíos que los impulsa a cambiar constantemente y tomar una conducta autónoma en determinadas situaciones.

Por otro lado, vemos que las organizaciones fácilmente pueden percibir los beneficios de los procesos (facilitan las revisiones de calidad, la inducción del personal, las auditorías, etc) siempre y cuando estos procesos se lleven a la práctica y no permanezcan como simples documentos, ya que por sí solos no traen valor a la organización. Esta premisa aplica tal cual, a los proyectos ágiles, pues cuando hablamos de procesos en los proyectos debemos mantener un enfoque hacia la práctica (que sean interiorizados y ejecutados naturalmente por todos los miembros), manteniendo un sano equilibrio entre la típica burocracia organizacional (que desfavorece la agilidad) y la anarquía (que disminuye la credibilidad en el equipo). Cuando se encuentra y mantiene el punto medio entre estas dos, el equipo se encuentra en un entorno de agilidad: sin burocracia absoluta y sin anarquía absoluta.



Ilustración 13 - Equilibrio entre burocracia y anarquía: agilidad

5.4- Centrado en el valor para cliente (4)

En un proyecto Scrum, la máxima prioridad es satisfacer al cliente desde el inicio y continuamente entregándole el máximo valor posible, para lo cual es importante considerar los siguientes aspectos:

- Es importante que en cada uno de los Sprints se generen incrementos de producto que entreguen valor para el cliente y estos a su vez estén "terminados".
- Para realizar la priorización de los elementos que hacen parte del Product Backlog los miembros del Equipo Scrum deberán considerar principalmente el valor que el elemento puede generar, para ello es importante el concepto de transparencia (*Sección 5.1.1*).
- Cada incremento de producto "terminado" debe ser validado con el cliente para asegurar la recolección de retroalimentación.
- Es altamente importante que el cliente participe activamente de las revisiones de los prototipos del producto antes de realizar cualquier desarrollo; incluso y según la naturaleza del proyecto, el cliente podrá participar activamente del diseño del producto.

5.5– Cumplimiento (5)

Cumplir con las reglas establecidas por el Equipo Scrum y por la organización es de vital importancia para garantizar una sana convivencia entre todos los miembros del Equipo Scrum, evitar desvíos en los proyectos y, por último, pero no menos importante, garantizar la satisfacción del cliente.

A continuación, se listan las reglas esenciales que deben ser cumplidas:

- Reglas establecidas por el Equipo Scrum.
- Reglas establecidas por la organización.
- Bloques de tiempo asignados a los eventos Scrum.
- Definición de terminado y criterios de aceptación.

5.5.1 - Bloques de tiempo

Los bloques de tiempo asignados a los eventos Scrum, garantizan que no se desperdicia tiempo en los proyectos. Algunas ventajas de respetar los Bloques de Tiempo asignado son los siguientes:

- Se evita que el equipo pierda motivación.
- Menos gastos generales para el proyecto.
- Se garantiza una alta velocidad para los equipos.
- Las prácticas relacionadas con el desarrollo de entregables son más eficientes.

5.5.2 - Reglas del Equipo

Gracias al principio de la autoorganización, son los miembros del Equipo Scrum, quienes establecen sus propias reglas, claro, considerando el cumplimiento de las reglas establecidas por la organización. Por lo general, estas reglas se establecen una única vez al inicio del proyecto, y en algunas ocasiones podrán ser globales para múltiples proyectos o múltiples equipos. Al artefacto donde se registran estas reglas suele llamarse “Plan de Colaboración del Equipo Scrum”.

Algunos de los ítems que pueden hacer parte del Plan de Colaboración del Equipo Scrum son:

- Principios del equipo.
- Herramientas de comunicación.
- Horarios de las reuniones.
- Penalizaciones por incumplimiento.

5.6- Desarrollo iterativo (6)

Scrum fue diseñado para que el desarrollo del proyecto se realice por iteraciones, también conocidas como Sprints. El método iterativo es flexible y abierto a los cambios, lo que permite adaptar el proyecto a las necesidades cambiantes del mercado, del cliente o de la organización.

Cada iteración está compuesta por las siguientes etapas del Ciclo de vida del Proyecto y sus respectivas prácticas:

- Etapa 2: Planificación.
- Etapa 3: Desarrollo del Sprint.
- Etapa 4: Revisión del Sprint.
- Etapa 5: Implementación.

Esto significa que las siguientes etapas tienen prácticas que no necesariamente son iterativas:

- Etapa 1: Inicio del proyecto.
- Etapa 6: Cierre del proyecto.

6. Consideraciones para la Gestión del Proyecto

6.1 - Contratación

6.1.1 - Roles del proyecto

Las personas ocupan un papel clave en la gestión de proyectos ágiles, su participación se da a través del conjunto de responsabilidades que les son asignadas, es decir, su rol en el proyecto. En un proyecto Scrum se consideran 2 categorías de clasificación para los roles:

6.1.1.1 - Comprometidos

Los **comprometidos** son los roles que **obligatoriamente** se requieren para producir el producto del proyecto, por ende, son los responsables del éxito de cada iteración del proyecto y del proyecto en sí.

Los roles comprometidos son:

- El Product Owner.
- El Scrum Master.
- El Equipo de Desarrollo.

6.1.1.2 - Involucrados

Los **involucrados** son los roles que no son obligatoriamente necesarios para el proyecto Scrum. Ellos pueden interactuar con el equipo, pero **NO** son responsables del éxito del proyecto.

Los roles involucrados son:

- Clientes.
- Usuarios.
- Proveedores (Ej: Freelancers; Proveedores de servicios; etc).

- Oficina de Gestión de Proyectos Ágiles (APMO).

6.1.2 - Contratación del equipo (Personal)

A continuación, se describen algunas de los requisitos que se deben considerar para la asignación/contratación de las personas que ocuparán los distintos roles en un proyecto Scrum.

Product Owner	Scrum Master	Equipo de Desarrollo
<ul style="list-style-type: none"> - Experto en Scrum - Amplia experiencia y dominio del negocio - Buen negociador - Organizado - Experiencia en dirección de proyectos - Líder servicial 	<ul style="list-style-type: none"> - Experto en Scrum - Capacidad para resolver problemas - Habilidades de coordinación - Líder servicial 	<ul style="list-style-type: none"> - Conocimiento básico de Scrum - Expertos técnicos - Multifuncionales - Proactivos

Ilustración 14 - Habilidades del Equipo Scrum

6.1.2.1 - Los líderes serviciales

Los líderes serviciales tienen varias características que les permiten apoyar al Equipo de Desarrollo:

- **Capacidad de escuchar:** Escuchan con atención y son receptivos a lo que se dice y no se dice para comprender y reflexionar sobre la situación.
- **Empatía:** Aceptan y reconocen a los individuos por sus habilidades y destrezas únicas.
- **Persuasión:** Logran el consenso del Equipo en la toma de decisiones.

6.1.2.2 - Matriz de competencias

La matriz de competencias le permite a la organización identificar las competencias necesarias en los miembros que hacen o harán parte de los distintos equipos y así encontrar cualquier brecha que pueda

existir en los miembros del Equipo Scrum, y así también poder identificar los miembros que necesitarán capacitación adicional en un área o competencia específica.

			Posibles miembros	
Habilidad/Competencia	Nivel Necesario	Costo de Capacitación	Juan	Maria
Scrum	Experto	\$1.000	Experto	Intermedio
Bases de Datos	Intermedio	\$1.800	Experto	Novato
Contabilidad	Novato	\$900	Novato	Nulo

Ilustración 15 – Matriz de competencias

Aunque generalmente para la contratación de los miembros del equipo se realiza con base en el conocimiento o habilidades técnicas, en Scrum se reconoce que también es necesario que los miembros del equipo cuenten con otro tipo de habilidades que le permitan desarrollar mejores interacciones, mejor comunicación y mayor cohesión. Las necesidades del equipo Scrum, se toman como criterios adicionales a tener en cuenta para la adecuada contratación de los miembros:



Ilustración 16 – Necesidades del equipo

6.1.2.3 - Roles del Equipo de Desarrollo

Se suele pensar que en los equipos multifuncionales no pueden existir roles ni especialistas, siendo esta afirmación totalmente falsa, por lo general un equipo está compuesto por distintos especialistas tales como:

- Analistas.
- Desarrolladores.
- Probadores.
- Integradores.
- Diseñadores.
- Arquitectos.
- Etc.

¿Contratar Novatos o Expertos?

Si el objetivo es terminar el proyecto en poco tiempo y con muy alta calidad deberás contratar expertos técnicos, considerando que esto tendrá un impacto significativo en el costo; por otro lado se podrían contratar solo novatos, pero el proyecto tardará bastante en desarrollarse debido a la curva de aprendizaje, además la calidad podría verse comprometida.

Realizar una combinación entre novatos y expertos, puede desmotivar a los expertos, aunque también puede darse la situación en la que los novatos crezcan rápido y se logre el equilibrio esperado.

6.1.3 - Tipos de contratos

Contrato de precio fijo

- Se define un precio total fijo para todo lo que se desarrolle en el proyecto.
- Mayor riesgo financiero si no se da cumplimiento a todo el proyecto en función del contrato.
- Los entregables son fijos y definidos al inicio del proyecto.
- **No son ideales para el uso de Scrum.**

Contrato de Unión Temporal

- Generalmente se utiliza cuando dos o más socios ejecutan un proyecto.
- El ROI (ingresos o beneficios) será compartido entre todos los socios.
- Es importante definir los % de participación y las responsabilidades en el proyecto.
- Es ideal centralizar los equipos o dividir el proyecto en componentes.

Contrato de desarrollo en fases

- Se considera el concepto de Producto Mínimo Viable.
- Se “fragmenta” el proyecto en varias fases, donde al final de cada fase se hacen pagos.
- Cada fase genera un conjunto importante de entregables.
- Útil para proyectos de gran tamaño.
- Se reduce el riesgo monetario, ya que los despliegues sin éxito no son financiados.

Contrato de entrega incremental

- El cliente/patrocinador puede tomar decisiones sobre el proyecto en cada inspección: Puede aceptar el entregable, Detener el desarrollo o Solicitar modificaciones.
- Cada Sprint debe generar un incremento por lo general es una épica o componente completo.
- El pago/facturación se hace con cada Iteración (se utiliza generalmente en proyectos internos).
- También se le llama “Contrato por Sprints”.

6.1.4 - Finanzas del proyecto

6.1.4.1 - Presupuesto inicial del proyecto

Una de las prácticas clave en Scrum es la construcción del Presupuesto inicial del proyecto, para lo cual se deben tener en cuenta como mínimo los siguientes elementos:

- Personas que harán parte del proyecto.
- Materiales.
- Servicios.
- Infraestructura.
- Capacitación del equipo.
- Reservas.
- Otros gastos que afecten el desarrollo del proyecto.

Nota: Es responsabilidad del Product Owner y el Patrocinador del proyecto, discutir, negociar y aceptar el presupuesto para asegurar que haya suficientes fondos disponibles para el proyecto.

Algunas de las técnicas y herramientas que se pueden utilizar para gestionar las finanzas del proyecto son:

6.1.4.1.1 - Retorno de la Inversión

$$\text{ROI} = \frac{\text{Ingresos del proyecto} - \text{Costo del proyecto}}{\text{Costo del proyecto}}$$

Ilustración 17 - Fórmula para el cálculo del ROI

6.1.4.1.2 - Estimación basada en históricos

Los datos históricos del mismo proyecto o de otros proyectos serán de gran utilidad para lograr mejores estimaciones de los elementos que hacen parte del Product Backlog.

Dependiendo de la información disponible en la organización, se puede utilizar:

- **Históricos reales de proyectos anteriores:** Esta técnica funcionará siempre y cuando la organización mantenga el registro del esfuerzo y duración reales del proyecto (la técnica más recomendable en Scrum).
- **Históricos planeados:** En los que se cuenta con un estimado original, aunque no se haya tenido un seguimiento a los datos reales (la técnica más común).

6.1.4.1.3 - Coeficiente de conocimiento del negocio

Esta técnica está basada en la idea de que a mayor conocimiento del negocio/proyecto, más precisa será la estimación del presupuesto. A menor conocimiento del negocio/proyecto, habrá mayor riesgo y menor precisión en la estimación.

El objetivo de esta técnica es calcular en una escala porcentual el grado de conocimiento sobre el negocio y a partir de allí calcular la reserva financiera o presupuestal.

Algunos factores para el cálculo del coeficiente de conocimiento sobre negocio son:

- Experiencia en proyectos similares.

- Calidad de requerimientos iniciales del proyecto.
- Grado de riesgo al que está expuesto el proyecto.
- Desviación (%) en las estimaciones de proyecto.

6.1.5 – Contratación de proveedores

Para garantizar que se trabaja con los mejores proveedores, la selección se realiza considerando las opiniones de todo el Equipo Scrum.

Algunos de los criterios que se pueden considerar para la selección de los proveedores son:

- Precio.
- Curva de Aprendizaje.
- Facilidad de uso.
- Velocidad de la Solución.
- Soporte.
- Seguridad.
- Complejidad para la migración de Datos.
- Tiempo de Implementación.
- Reputación.
- Facilidad de Pago.
- Referencias.

6.1.6 – Los “Freelancers”

En algunos proyectos, podría ser necesario el apoyo de personal no disponible en el Equipo de Desarrollo, por ejemplo “Arquitectos”, “Diseñadores”, “Administradores de Bases de Datos”, etc.

Este tipo de personal es considerado como proveedor, y normalmente solo se contrata por días, y en momentos muy específicos del proyecto.

*Ilustración 18 - Freelancers*

6.2 - Seguimiento y control del proyecto

El seguimiento y control del proyecto, es una actividad que debe hacerse de forma continua a lo largo del ciclo de vida del proyecto. Por lo general los momentos donde se realiza de forma intensiva son: Al inicio, a intervalos predefinidos durante el proyecto o en cualquier momento cuando surgen problemas o riesgos de viabilidad.

6.2.1 - Oficina de Gestión de Proyectos Ágiles (APMO)

Una APMO está conformada por un Grupo de expertos en Scrum u otros Marcos de Trabajo ágiles.

La Oficina de Gestión de Proyectos Ágiles (APMO) tiene como responsabilidades:

- Estudiar la viabilidad de las iniciativas de proyectos.
- Generar guías de apoyo.
- Brindar asesoría a los diferentes Equipos Scrum de la Organización.
- Dar seguimiento a los proyectos.
- Reportar a la Alta Dirección.
- Mantener actualizado el portafolio de proyectos.
- Definir los "Criterios de terminado" globales para la organización (*Ver sección - Definición de terminado (DoD)*).
- Por lo general se encarga de guiar las retrospectivas de proyecto.
- Garantizar la correcta gestión de las lecciones aprendidas de los proyectos.
- No toma decisiones sobre los proyectos, pero funciona como apoyo de consultoría, asesoramiento u orientación para todos los proyectos, programas y portafolios de proyectos de la organización.

*Ilustración 19 - Actividades de la APMO*

6.2.2 - Informes y reportes del proyecto

En cualquier momento es posible conocer el progreso del proyecto tan solo con sumar el trabajo total restante para alcanzar el objetivo. El Product Owner hace seguimiento de este trabajo restante total al menos en cada Revisión de Sprint. El Product Owner compara esta cantidad con el trabajo restante en Revisiones de Sprint previas, para evaluar el progreso hacia la finalización del trabajo proyectado en el tiempo deseado para el objetivo. Esta información se muestra de forma transparente a todos los interesados.

Varias prácticas de proyección de tendencias se han utilizado para predecir el progreso, como trabajo pendiente (Burn Down), trabajo completado (Burn Up) y el flujo acumulado (Cumulative Flow). Estas han probado ser útiles, sin embargo, no reemplazan la importancia del empirismo. En entornos complejos se desconoce lo que ocurrirá. Solo lo que ya ha ocurrido puede utilizarse para la toma de decisiones con miras al futuro.

6.2.2.1 - Velocidad del Equipo Scrum

La velocidad del Equipo es la velocidad con la que el equipo puede completar el trabajo en un Sprint. Por lo general se expresa en las mismas unidades que las utilizadas para la estimación, normalmente puntos de historia.

- Todos los equipos tienen diferente velocidad, así hagan parte del mismo proyecto.

A continuación, la fórmula para calcular la velocidad de un equipo Scrum:

$$\text{Velocidad} = \frac{\text{Puntos de Historia a desarrollar}}{\text{Número de Semanas del Sprint} \times \text{Cantidad de personas en el equipo}}$$

Velocidad = 84 Puntos
 2 Semanas
 7 Integrantes

Ilustración 20 - Velocidad del Equipo Scrum

6.2.2.2 – Métricas en proyectos Scrum

Algunas de las métricas que son realmente útiles cuando se trabaja usando Scrum son:

Métrica	¿Qué se mide?	¿Para qué se utiliza?	Responsable	Frecuencia
Flujo acumulado	Historias de Usuario Completadas	<ul style="list-style-type: none"> • Monitorear el progreso del equipo. • Evitar la deuda técnica del equipo en cada sprint. 	Scrum Master	Sprint
	Historias de Usuario Pendientes			
	Historias de Usuario En Progreso			
Costos de Proyecto	Costos mensuales del Proyecto	Disminuir el riesgo de desvíos financieros.	Product Owner	Mes
Velocidad de Equipo	Cantidad de puntos de historia que el equipo puede terminar en un Sprint	Encontrar el punto de equilibrio del trabajo al que se puede comprometer el equipo en cada sprint.	Scrum Master	Sprint

Impedimentos	Cantidad de Impedimentos Resueltos	Evitar que el equipo tenga inconvenientes o atrasos en el próximo sprint.	Equipo de Desarrollo / Scrum Master	Sprint
Cambios	# Cambios aceptados	Garantizar la estabilidad de los proyectos y Garantizar que los cambios son atendidos a tiempo.	Product Owner	Semanal
	# Cambios rechazados			
	# Cambios pendientes por revisar			
Errores	Errores en pruebas unitarias	Garantizar la calidad del producto.	Equipo de Desarrollo	Sprint
	Errores en pruebas de colegas			
	Errores en producción			
	Errores en Validación			
Riesgos	Riesgos Materializados	Garantizar la calidad del producto y la estabilidad del proyecto	Product Owner	Sprint
	Riesgos Mitigados			
	Riesgos No tratados			

6.2.3 - Seguimiento de Calidad

En Scrum, la calidad se define como la capacidad del producto para cumplir criterios de “terminado” y alcanzar el valor que espera el cliente. En los proyectos Scrum es sumamente importante realizar un seguimiento constante a la calidad del producto y así evitar inconvenientes en el futuro (esta es una práctica iterativa que se realiza en todos los Sprints).

6.2.3.1 - Control de calidad

El Control de Calidad se refiere a la ejecución de las actividades de calidad que se realizan a los incrementos de producto que están potencialmente listos para la entrega y posteriormente sobre el Producto.

Normalmente los controles de calidad son realizados por el Equipo de Desarrollo y el Scrum Master durante la práctica de Desarrollo del Sprint, y por el Product Owner en la Revisión del Sprint.

6.2.3.2 - Aseguramiento de calidad

El Aseguramiento de la Calidad se refiere a la evaluación de los procesos y normas que están definidos en el Ciclo de vida de un proyecto Scrum.

Normalmente el aseguramiento de calidad se hace por medio de auditorías de proceso, realizadas por la APMO (Sección 6.2.1).

Para realizar las actividades de Aseguramiento de calidad, puede usarse el Marco para la Evaluación y Mejora de Procesos de Tecnologías de la Información (MEMPTI).

6.3 - Gestión de Riesgos

La gestión de riesgos es una actividad que se realiza proactivamente y a través del ciclo de vida del proyecto.

- Un riesgo es definido como una situación inesperada que puede afectar los objetivos de un proyecto.
- Los riesgos pueden tener un impacto tanto positivo como negativo sobre el proyecto.

6.3.1 - Ejemplos de riesgos comunes:

- **Falta de capacidad del recurso humano actual**

En algunas organizaciones, el personal disponible para la ejecución de los proyectos tiene asignadas otras responsabilidades que le impiden tener total disponibilidad para ejecutar las actividades del proyecto.

Algunas de las posibles acciones a tomar son:

- Planificar las actividades del personal actual.
- Evitar en lo posible los cambios urgentes.
- Considerar la contratación de personal o de proveedores externos que puedan aportar a la ejecución de los proyectos.

- **Falta de seguimiento y control sobre los proyectos**

La falta de seguimiento y control del portafolio de proyectos puede provocar su obsolescencia y/o fracaso.

Algunas de las posibles acciones a tomar son:

- Usar una herramienta para el seguimiento y control de los proyectos.
- Realizar una actualización continua del portafolio de proyectos, para confirmar la entrega de valor.
- Monitorear el cronograma y el presupuesto de cada proyecto.

- **El presupuesto se agota o no es suficiente**

Se debe considerar que, para el desarrollo de la mayor parte de los proyectos internos, se necesitan considerables sumas de dinero, que deberán ser proyectadas dentro del presupuesto general de la organización.

- **Falta de compromiso por parte de los involucrados en los proyectos**

Los proyectos Scrum requieren una colaboración entre los miembros del equipo y las partes interesadas, por lo que es importante contar con el compromiso de todos los involucrados.

Para este fin, se recomienda:

- Definir claras y correctas expectativas para todos los involucrados.

- Comunicar a tiempo y con el detalle suficiente los compromisos e importancia de la participación de los involucrados.
- Mantener siempre informados a los involucrados sobre los avances y/o cambios que puedan tener los proyectos (utilizar un lenguaje sencillo que todos los involucrados puedan entender).
- Evitar reuniones que no generen valor o reuniones demasiado extensas que interrumpen significativamente las actividades de los involucrados.
- Comunicar siempre los resultados y beneficios logrados con la ejecución de cada proyecto.
- Agradecer al equipo involucrado y funcionarios su participación y resaltar la importancia de sus ideas y revisiones.

- **Posible rotación del personal actualmente involucrado en el proyecto**

Uno de los mayores riesgos a los que normalmente se encuentra expuesto un proyecto, es la posible renuncia de los miembros del equipo, por lo que se recomienda:

- Realizar transferencia de conocimiento entre los miembros del equipo.
- Constituir una base de conocimiento y lecciones aprendidas.
- Garantizar la documentación continua.

6.3.2 - Apetito de Riesgo

El Apetito de Riesgo es un modelo utilizado para medir la preferencia de las Partes Interesadas por el riesgo o su actitud hacia el riesgo. Esto define el nivel de las Partes Interesadas para aceptar riesgos.

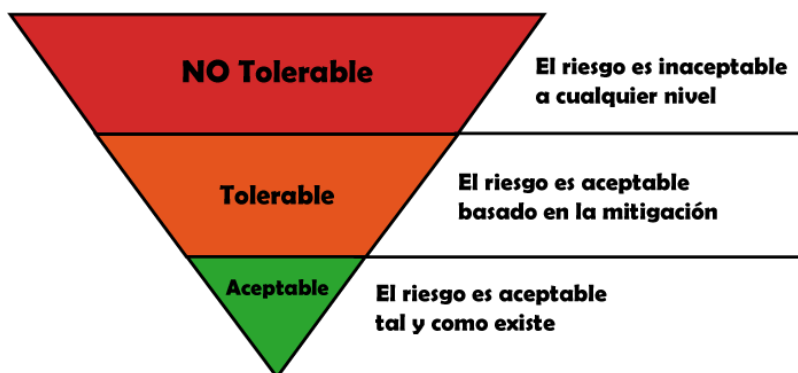


Ilustración 21 - Apetito de riesgo

6.3.3 - Tolerancia al Riesgo

La tolerancia al riesgo es la cantidad máxima de riesgo que la organización está dispuesta a aceptar para lograr los objetivos del proyecto; la tolerancia al riesgo sirve como una alerta para evitar llegar a la capacidad de riesgo.

Capacidad de Riesgo

La capacidad de riesgo es el nivel de riesgo máximo que se puede permitir antes de que el proyecto se desvíe de tal forma que no entregue valor al cliente. En caso de superar la capacidad de riesgo, por lo general se da por terminado el proyecto (*Sección - Etapa 6: Cierre del proyecto*).

6.3.4 - Ciclo de vida de la gestión de riesgos

La gestión de riesgos se compone de cinco pasos que se enuncian a continuación:



Ilustración 22 - Ciclo de vida de la gestión de riesgos

6.3.4.1 - Identificación del riesgo (1)

La identificación de riesgos permite conocer los posibles riesgos que pueden afectar el desarrollo del proyecto y de sus respectivos Sprints.

Existen 2 momentos importantes donde se realiza la identificación de riesgos:

- **Al inicio del proyecto:** Se identifican los riesgos globales del proyecto, por ejemplo, riesgos relacionados con el presupuesto, el personal, etc.
- **Los Sprints:** Se identifican los riesgos que pueden afectar el desarrollo de ese Sprint particular, por ejemplo, riesgos del producto. Esta actividad se realiza de forma iterativa durante todo el proyecto principalmente en las reuniones de Planificación de los Sprint.

Solo mirando el proyecto desde diferentes perspectivas y utilizando una variedad de técnicas, se puede hacer la identificación de los posibles riesgos. La técnica más utilizada es la lluvia de ideas.

6.3.4.2 - Evaluación del riesgo (2)

La evaluación de riesgos ayuda a entender el impacto potencial de un riesgo, ¿qué tan probable es que se produzca, y cuándo es posible que el riesgo se materialice? Con ello una decisión podrá ser tomada y determinar si sería buena idea continuar con el Sprint o incluso el Proyecto.

La evaluación de riesgos se hace considerando 3 factores:

- **Proximidad** = Cantidad de días o fecha estimada en la que se podría presentar el riesgo.
- **Probabilidad** = Medida porcentual que ayuda a determinar la posibilidad de que el riesgo ocurra.
- **Impacto** = Mide el daño que ocasiona el riesgo, suele clasificarse numéricamente según las siguientes categorías: Crítico (6), Muy Alto (5), Alto (4), Medio (3), Bajo (2), Muy Bajo (1)



Ilustración 23 - Criterios para la evaluación de riesgos

6.3.4.2.1 – Valor Monetario Esperado

Esta técnica se utiliza para calcular el impacto monetario que podrá tener un riesgo, y con esta información realizar reservar monetarias para la prevención y/o mitigación de riesgos.

La técnica considera 2 factores: El impacto monetario del riesgo y la probabilidad de ocurrencia.

$$\text{VME} = \text{Impacto del Riesgo (Dinero)} \times \text{Probabilidad de Riesgo (\%)}$$

Ejemplo: Una organización que implementa sistemas de energía solares detecta una posible tormenta tropical que le impediría continuar con la instalación de los paneles previstos para el próximo Sprint. Para calcular el impacto que esto tendría en el proyecto se identifican factores como posibles multas, salario de las personas, etc. Llegando a la conclusión de que para este riesgo se perderían 538 dólares. El siguiente paso es calcular la probabilidad de ocurrencia del riesgo, que según el instituto meteorológico es del 35%. Al aplicar la técnica como se explica, el Valor Monetario de ese riesgo es de 188.3 dólares.

$$\begin{array}{ccccc}
 \$538 & \times & 35\% & = & 188,3\$ \\
 \text{(Impacto)} & & \text{(Probabilidad)} & & \text{VME}
 \end{array}$$

6.3.4.3 - Priorización del riesgo (3)

La priorización de riesgos permite establecer un orden para la mitigación de los riesgos. Para realizar la priorización de los riesgos se siguen los siguientes pasos:

- Identificar la fecha aproximada en la que se presentaría el riesgo. (Considerar que los riesgos más próximos deben ser atendidos primero).
- Calcular el factor de exposición (Relación entre probabilidad e impacto) y con este factor, prioriza entre los riesgos más próximos.

$$\begin{array}{ccccc}
 3 & \times & 48\% & = & 1,44 \\
 \text{Impacto} & & \text{(Probabilidad)} & & \text{Factor de} \\
 \text{Medio} & & & & \text{exposición}
 \end{array}$$

6.3.4.4 - Mitigación del riesgo (4)

En la etapa de mitigación, el Equipo Scrum determina la acción a tomar con el riesgo. En Scrum existen 3 posibles respuestas a los riesgos:



Ilustración 24 - Acciones de mitigación de riesgo

- La respuesta a cada riesgo dependerá de la probabilidad y el impacto del riesgo.
- La naturaleza iterativa de Scrum, con sus ciclos de tiempo de respuesta y retroalimentación rápida permite que las fallas se detecten de forma temprana; por lo tanto, hablando en términos prácticos, tiene una función de mitigación natural construida dentro del sistema.
- Los riesgos pueden ser mitigados mediante la implementación de una serie de respuestas que pueden ser Proactivas/preventivas o reactivas.

6.3.4.4.1 – Mitigación

- La mitigación se enfoca en reducir la probabilidad de ocurrencia o el impacto del riesgo.
- Comprende acciones que se toman por adelantado o acciones Proactivas.
- Se reduce la exposición al riesgo (probabilidad vs impacto) dentro de límites aceptables para el proyecto.

6.3.4.4.2 – Contingencia

- La contingencia se enfoca en definir una respuesta que se utiliza si el riesgo se materializa o se identifican señales de advertencia.
- Comprende acciones Reactivas y acciones de monitoreo en caso de que el riesgo sea inevitable.

6.3.4.5 - Comunicación del riesgo (5)

Las Partes Interesadas deben ser informadas continuamente acerca del estado de los riesgos, incluyendo el impacto potencial de estos riesgos y los planes para responder a cada riesgo.

Por lo general, la comunicación del Riesgo es llevada a cabo por el Scrum Master hacia el Product Owner y por el Product Owner hacia el cliente.

Esta comunicación siempre está en curso y debe ocurrir en paralelo durante los cuatro pasos secuenciales discutidos hasta ahora.

6.4- Gestión de Cambios

Ágil implica la apertura al cambio, por lo que es común que todos los proyectos ágiles estén expuestos a cambios, y es de vital importancia que los miembros del equipo estén preparados para enfrentar los cambios en cualquier etapa del proyecto.

Es aún más importante que la organización sea consciente de la exposición a los cambios para crear sinergia con los equipos Scrum y buscar aprovechar los beneficios minimizando el impacto negativo que pudiese resultar del cambio.

Dada la naturaleza iterativa o incremental de Scrum, es posible manejar los cambios sobre el producto de una manera ordenada y cíclica, respondiendo continuamente a lo que espera el cliente; en correspondencia con el manifiesto ágil, *“Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente”*.

Aunque en Scrum hay una fuerte inclinación a no mantener grandes cantidades de información sin valor, una forma apropiada de solicitar los cambios sobre los productos Scrum, es a través del formato para Solicitud de Cambios – RFC (Request For Change), sin embargo esto no implica que necesariamente haya documentación de por medio.

Un buen RFC, debería tener como mínimo los siguientes componentes:

- ¿Quién solicita el cambio? ¿Por qué? ¿Para qué necesitamos hacerlo?
- ¿Para cuándo se necesita?
- ¿Qué tan importante es?
- ¿Qué pasa si no se realizara/aprobara el cambio?
- ¿Quién lo va a ejecutar? ¿Qué necesita? ¿Es un empleado o debemos contratar?

Es importante que detrás de cada cambio solicitado exista un “iniciador” para mantener la traza entre el cambio y saber de dónde proviene. Además dentro de la gestión de cambios en Scrum se reconoce la autoridad del Product Owner en la aprobación/rechazo de los cambios, por lo que es importante que durante todo el proyecto, el Product Owner esté enterado de lo que sucede con el producto.

A continuación, se describe el flujo de los cambios:

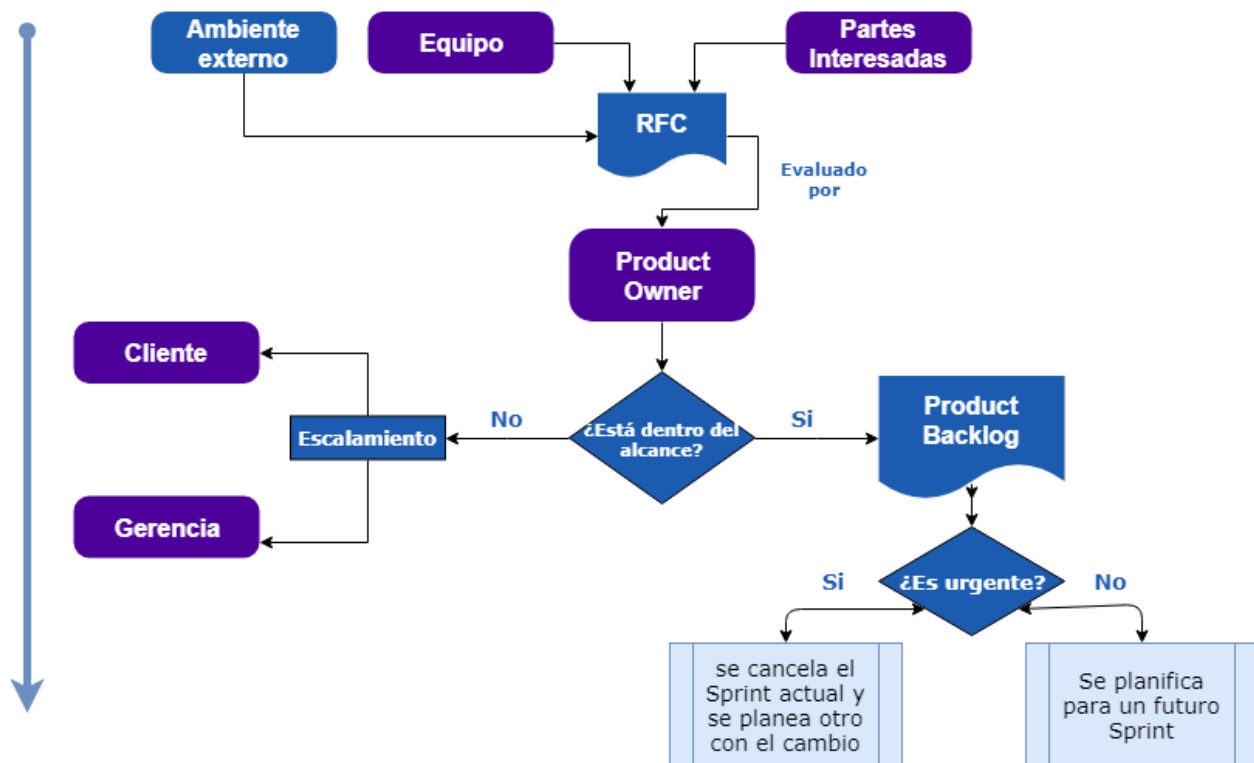


Ilustración 25 - Flujo de cambios

- El Product Owner es el rol responsable de aprobar/rechazar los cambios, sin embargo, en algunas ocasiones cuando los cambios se salen de su conocimiento o están por fuera del alcance definido para el proyecto podrá escalar los cambios a la gerencia de la organización.
- Los cambios pueden venir de distintas fuentes, las más comunes son:
 - Cambios solicitados por las partes interesadas.
 - Solicitudes realizadas por los miembros del Equipo.
 - Nuevas tecnologías emergen y es necesario realizar cambios al producto.

- El producto definido comienza a perder vigencia y es necesario cambiar el alcance.

6.5- Diseño del producto

Scrum es usado comúnmente para la Gestión de Proyectos ágiles orientados a la construcción de productos/servicios, sin embargo, etapas previas al proyecto como la concepción, prototipado y diseño del producto no siempre se explican al detalle, siendo altamente importantes para el buen desarrollo del proyecto.

Algunos de los conceptos que se deben considerar en esta etapa son:

6.5.1 - Producto mínimo viable

El definir el Producto Mínimo Viable o también llamado Características Mínimas de Mercado es una actividad extremadamente importante, de modo que la primera versión del producto se construye tan pronto como sea posible, lo que lleva a un aumento de rendimiento de la inversión.

Normalmente, estos requerimientos se ubicarían como alta prioridad dentro del Product Backlog.

El Producto Mínimo Viable se define entre el Cliente y el Product Owner.

6.5.2 - Análisis de viabilidad del producto

El objetivo del modelo Lean Canvas (adaptado de Business Model Canvas) es el de identificar la viabilidad de un producto o servicio y así disminuir el riesgo y los posibles obstáculos.

En muchas ocasiones el Lean Canvas se utiliza como el artefacto sustituto del caso de negocio.

Las claves a la hora de construir un Lean Canvas son:

- **Crear:** Plantear la idea y probar con prototipos de bajo costo.
- **Medir:** Comprobar el interés de posibles usuarios y medir resultados.
- **Aprender:** Con los resultados se decide si se continua con la idea o se cambia algo.

El Lean Canvas está compuesto por 9 cuadrantes, tal como los que se describen a continuación:



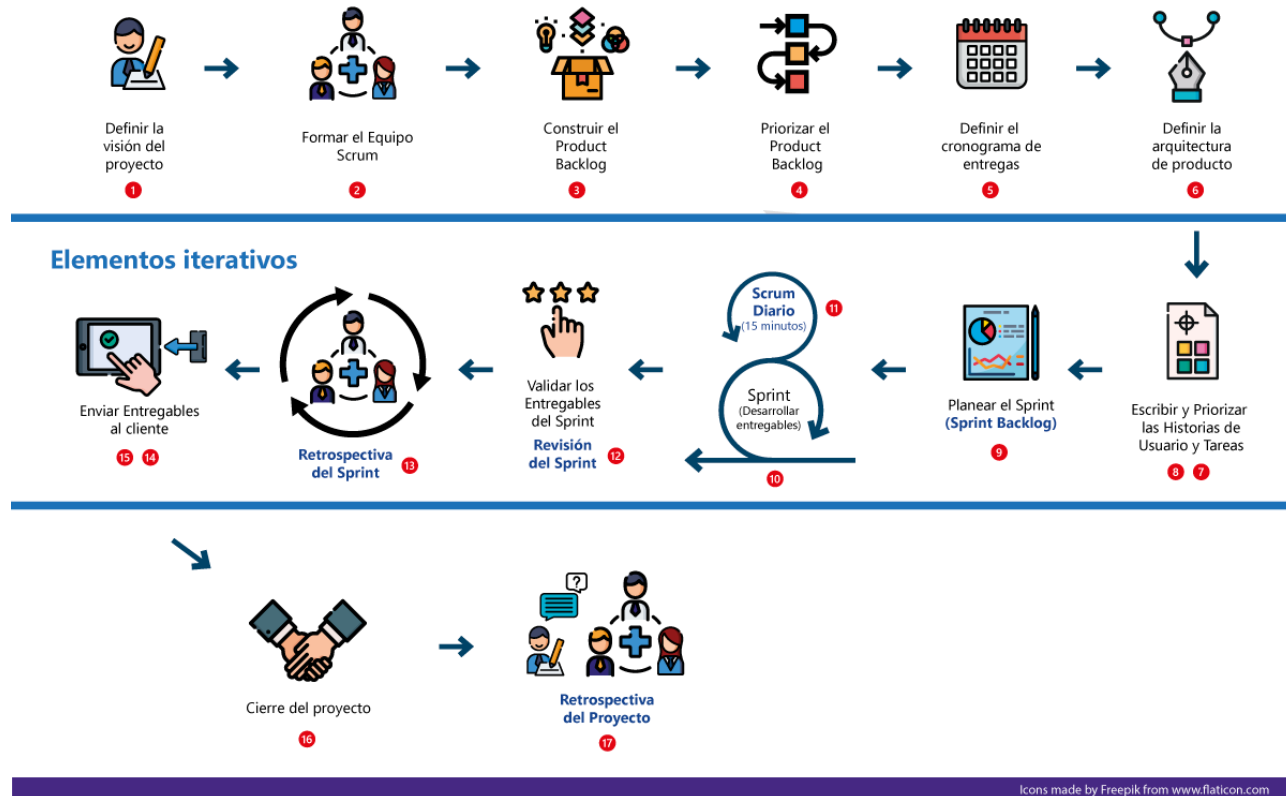
Ilustración 26 - Lean Canvas

7. El Ciclo de Vida en los Proyectos Ágiles

Scrum fue diseñado como un Marco de Trabajo basado en la simplicidad, por lo que no utilizamos el concepto “proceso” para describir las prácticas Scrum que permiten el desarrollo de un proyecto. Los procesos tal como se conocen es mejor dejarlos a las metodologías tradicionales.

Un proyecto desarrollado con Scrum tiene 6 etapas definidas en su ciclo de vida, cada etapa con un respectivo conjunto de prácticas (17 en total). A continuación, se muestra el ciclo de vida de un proyecto Scrum.

Ciclo de vida en un proyecto Scrum



Icons made by Freepik from www.flaticon.com

Ilustración 27 - Ciclo de vida en un proyecto Scrum

7.1 - Etapa 1: Inicio del proyecto

ID	Práctica	Rol vs Nivel de Involucramiento
Etapa 1: Inicio del proyecto		
1	7.1.1 - Definir la visión del proyecto (1)	<ul style="list-style-type: none"> Product Owner: Alto Scrum Master: Nulo Equipo de desarrollo: Nulo

2	7.1.2 - Formar el Equipo Scrum (2)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Alto • Equipo de desarrollo: N/A
3	7.1.3 - Construir el Product Backlog (3)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Medio • Equipo de desarrollo: Medio
4	7.1.4 - Priorizar el Backlog (4)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Medio • Equipo de desarrollo: Medio
5	7.1.5 - Definir el cronograma de entregas (5)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Alto • Equipo de desarrollo: Alto
6	7.1.6 - Definir la Arquitectura de Producto (6)	<ul style="list-style-type: none"> • Product Owner: Medio • Scrum Master: Medio • Equipo de desarrollo: Alto

7.1.1 - Definir la visión del proyecto (1)

En la práctica de "Definir la visión del proyecto" se estructura la visión del proyecto, explicando las necesidades empresariales que el proyecto busca satisfacer, considerando el alcance, el tiempo, el presupuesto y la calidad esperada por el patrocinador del proyecto.

Algunas de las actividades clave dentro de esta práctica son:

7.1.1.1 - Restricciones de un proyecto

Los proyectos siempre tienen limitaciones, también llamadas "restricciones". Por lo general se contemplan 4 restricciones principales (tiempo, presupuesto, alcance y calidad), sin embargo, según la naturaleza del proyecto y/o el estilo de la organización, podrían existir otras restricciones.

*Ilustración 28 - Restricciones del Proyecto*

7.1.1.2 - Identificar las partes interesadas

Al inicio del proyecto, el Product Owner, se encarga de garantizar que todas las Partes Interesadas son identificadas. Para ello, se recomienda que todas a su vez sean registradas, en lo que comúnmente se llama "Matriz de Partes Interesadas"

Esta matriz contiene la siguiente información:

- Empleados que trabajan para el proyecto, su respectivo rol y el tiempo en horas que dedican al proyecto.
- Proveedores del proyecto.
- Clientes / Usuarios que serán entrevistados, revisarán el proyecto o brindarán información útil para la construcción del producto.

7.1.1.3 - Definición de terminado (DoD)

Cuando un elemento del Product Backlog o del Incremento se describe como "Terminado", todo el mundo debe entender lo que significa "Terminado" (Done); aunque esto puede variar significativamente para cada Equipo Scrum.

Los miembros del equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado para asegurar la transparencia. Esta es la definición de "Terminado" ("Done") para el Equipo Scrum, y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos del Product Backlog puede seleccionar durante la Planificación del Sprint. El propósito de cada Sprint es entregar Incrementos de funcionalidad que potencialmente se puedan poner en producción y que se ajusten a la Definición de “Terminado” (Done) actual del Equipo Scrum.

Los Equipos de Desarrollo (Development Team) entregan un Incremento de funcionalidad de producto en cada Sprint. Este Incremento es utilizable, de modo que el Product Owner podría elegir liberarlo inmediatamente. Si la definición de “Terminado” (Done) para un incremento es parte de las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum (Scrum Team) deben seguirla. Si “Terminado” (“Done”) para un incremento no es una convención de la organización, el Equipo de Desarrollo del Equipo Scrum debe especificar una definición de “Terminado” (Done) apropiada para el producto. Si hay múltiples Equipos Scrum (Scrum Teams) trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum (Scrum Teams) deben definir en conjunto la definición de “Terminado” (Done).

Cada Incremento se integra con todos los Incrementos anteriores y es probado de manera exhaustiva, asegurando que todos los Incrementos funcionan en conjunto.

A medida que los Equipos Scrum maduran, se espera que su definición de “Terminado” (Done) se amplíe para incluir criterios más rigurosos para una mayor calidad. El uso de los nuevos criterios puede descubrir trabajo por hacer en los incrementos previamente “Terminados” (Done). Cualquier producto o sistema debería tener una definición de “Terminado” (Done) que es un estándar para cualquier trabajo realizado sobre él.

La Definición de Terminado puede ser definida para:

- Un sólo Sprint.
- Un Conjunto de Sprints.
- Un Proyecto.
- Un Programa.
- Toda la Organización.

Un ejemplo genérico de criterios de terminado puede ser:

- Los elementos fueron revisados por otros miembros del equipo (se realizaron las pruebas de colegas).
- Todos los defectos están arreglados.
- Se completaron las pruebas unitarias de la Historia de Usuario.
- Se finalizó toda la documentación técnica y de usuario final.
- Todos los archivos y documentos del proyecto están en el repositorio de la organización.
- Se hizo la demostración exitosa a los Socios y/o representantes de la empresa.
- El cliente dio su visto bueno a los entregables.

7.1.2 - Formar el Equipo Scrum (2)

Durante esta práctica se eligen los miembros que harán parte de los distintos Equipos Scrum.

7.1.2.1 - Plan de colaboración

Este artefacto define cómo las distintas partes interesadas y miembros del equipo de proyecto participan, se comunican y colaboran durante todo el proyecto. También se pueden definir las herramientas o técnicas específicas que se utilizarán para este fin.

Por ejemplo, cuándo y cómo se llevarán a cabo las reuniones, qué tipo de herramientas de comunicación se utilizarán, y quién debe estar involucrado en las diversas reuniones.

En proyectos grandes y complejos, especialmente con equipos distribuidos, este artefacto tiene mucha más formalidad. En otros proyectos pequeños, a veces puede ser simplemente un entendimiento verbal.

Por lo general es un artefacto que se construye en una reunión breve (30 minutos máximo), y se tiene en cuenta la opinión del todo el equipo.

7.1.2.2 - Reunión de inicio de proyecto o Kickoff

En esta reunión participan los responsables de ejecutar el proyecto y el cliente para formalizar el nuevo proyecto y en qué momento inicia. Esta reunión también resulta una herramienta que facilita la comunicación en el proyecto. (Esta reunión la dirige el Product Owner).

Los principales objetivos de la reunión de Kickoff son:

- Presentar los objetivos, beneficios y alcance del proyecto.
- Presentar el equipo de proyecto.
- Obtener el compromiso del equipo y partes interesadas frente al proyecto.
- Presentar el plan de trabajo inicial.
- Presentar la identificación inicial de riesgos.
- Definir la comunicación durante el proyecto.

7.1.2.3 - Modelo de desarrollo de equipos – Dr. Bruce Tuckman

El Dr. Bruce Tuckman definió el modelo para describir el curso que la mayoría de los equipos siguen en su camino hacia un alto rendimiento.

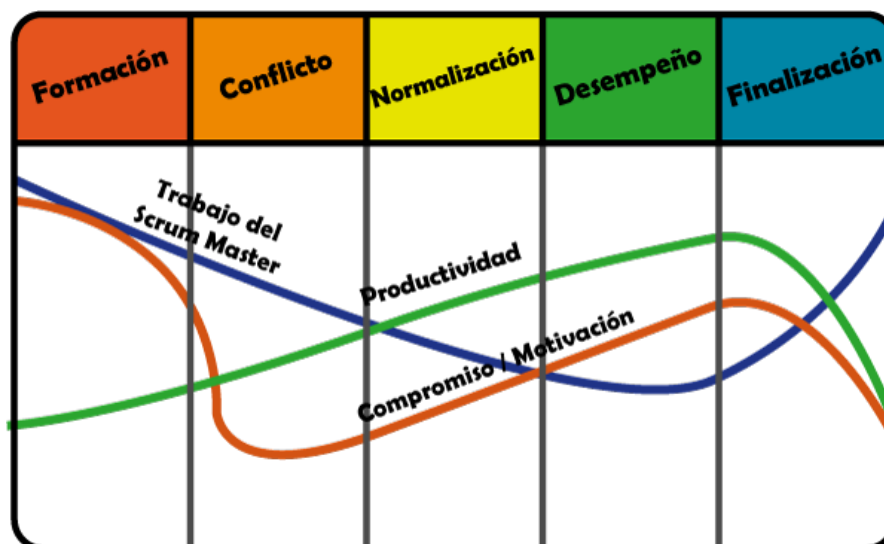


Ilustración 29 - Modelo de desarrollo de equipos – Dr. Bruce Tuckman

- **1. Formación (Dirección)**

Esta es la etapa en que se conforma el equipo y los miembros del equipo están empezando a conocerse, por lo que inicialmente su comportamiento tiende a ser individualista. Durante esta fase de Formación se establecen las primeras interacciones entre ellos, por lo que es común que no se presenten conflictos o discusiones.

En esta etapa también se definen las reglas del equipo, su propósito y su identidad (un nombre para el equipo), lo cual hace parte de un buen entorno de trabajo para el equipo.

En esta fase el Scrum Master toma una posición de “director” dado que los miembros del equipo dependen de él para la definición del rumbo del equipo y su orientación.

- **2. Enfrentamiento/Conflicto**

En esta fase ya se ha generado confianza entre los miembros del equipo, por lo que se presentan diferencias en el equipo y los miembros compiten entre sí, estableciendo o rompiendo relaciones entre ellos. Dado que cada miembro tiene su propia personalidad y estilo de trabajo, será necesario que se alcance un acuerdo de convivencia para evitar la disminución de la motivación.

Aunque la fase hace alusión al enfrentamiento, también es necesario que esos enfrentamientos se resuelvan para asegurar el buen rendimiento del equipo.

En esta fase el Scrum Master toma una posición de “coach”, brindando acompañamiento al equipo y guiando a los miembros para mantener los conflictos bajo control. También debe mantener la calma y dar ejemplo del comportamiento deseado para que los demás miembros del equipo lo imiten y adopten sanamente. El Scrum Master también es responsable de promover actividades que aumenten la motivación del equipo e identificar cuáles son las interacciones que mejoran o deterioran la convivencia en el equipo.

- **3. Normalización**

En esta fase los conflictos se reducen y se generan acuerdos que mejoran las interacciones entre los miembros del equipo. Durante esta fase los miembros comprenden mejor cuáles son sus responsabilidades, fortalezas y debilidades, lo cual propicia un entorno de mayor confianza y colaboración.

Cada miembro ya es consciente de su lugar en el equipo y es capaz de entender y aceptar el estilo de trabajo de los demás miembros, por lo que los intereses personales pasan a un segundo plano para dar prioridad a los intereses del equipo como un todo.

En esta fase el Scrum Master es un “facilitador”, ya que toma un enfoque hacia la mejora y acondicionamiento del entorno para facilitar el rendimiento del equipo. Dado que en esta fase el Scrum Master conoce más al equipo, puede identificar de qué manera encajan mejor sus miembros y cómo puede aumentar su rendimiento y motivación.

Es común que en esta fase se puedan incorporar nuevos miembros al equipo, por lo que el Scrum Master tiene que velar por mantener la estabilidad del equipo asegurando que los nuevos miembros se integren fácilmente al equipo sin disminuir su rendimiento ni su motivación.

- **4. Desempeño**

En esta fase se cuenta con un equipo maduro con suficiente confianza, motivación y autonomía por lo que los miembros están en la capacidad de tomar ciertas decisiones sin la presencia de un “jefe”, y se le pueden delegar tareas que antes eran exclusivas del Scrum Master (como llevar a cabo reuniones diarias, identificar impedimentos, etc.). Además, los miembros del equipo ya son capaces de resolver los desacuerdos o diferencias positivamente y en poco tiempo.

Se debe reconocer que no todos los equipos llegan a esta etapa pues se logra después de enfrentar conflictos y grandes dificultades, convivir y compartir experiencias, lo que implica que muchos equipos no logren completamente la etapa de Normalización y se queden estancados en la etapa de Enfrentamiento.

En esta etapa no se requiere de mucho esfuerzo por parte del Scrum Master, lo que le permite enfocarse mucho más en mantener y mejorar el entorno de trabajo del equipo favoreciendo el alto rendimiento de los miembros.

- **5. Finalización/Disolución**

Esta etapa comprende dos situaciones: la disolución del equipo o la finalización del equipo. La disolución se presenta cuando eventualmente durante el proyecto, uno o más integrantes se van del equipo, lo que provoca un atraso en el avance que ya se había logrado, conllevando a una disminución de la motivación y del rendimiento, pues el equipo queda resentido tras la pérdida de su estabilidad.

La finalización del equipo se presenta cuando acaba el proyecto y los miembros del equipo pasan a ser parte de otro proyecto o de otra empresa, lo cual implica que, aunque se haya alcanzado exitosamente el objetivo del proyecto, entre los miembros se genera una sensación de “pérdida” pues las interacciones que ya estaban establecidas y que ya funcionaban adecuadamente se terminarán junto con el proyecto. En algunos casos el mismo equipo puede hacer parte de un próximo proyecto, sin embargo, al tratarse de un nuevo proyecto no necesariamente significa que las interacciones permanecerán de la misma manera.

Es necesario que el Scrum Master brinde acompañamiento al equipo, pues ya sea si se trata de la disolución o la finalización, representa un cambio significativo para los miembros del equipo.

7.1.2.4 - Equipos de alto rendimiento

Un equipo de alto rendimiento es aquel que tiene un desempeño notablemente superior al promedio. En Scrum normalmente el rendimiento se asocia a la velocidad con la que el equipo consigue entregar valor. Para lograr equipos de alto rendimiento, se necesitan considerar como mínimo los siguientes factores:

- Se debe medir el rendimiento continuamente.
- Se deberían utilizar herramientas para automatizar la mayor cantidad de tareas posibles.
- La visión y reglas deben estar claras desde el inicio del proyecto.
- Se debe garantizar la presencia de elementos que fomenten la creatividad.
- Los objetivos definidos deben suponer un reto para el equipo.
- El equipo debe mantenerse motivado.

7.1.3 - Construir el Product Backlog (3)

Durante esta práctica se construye el Product Backlog (*Ver Sección 4.1 - Product Backlog*).

Los elementos que hacen parte del Product Backlog son:

- Épicas.
- Historias de Usuario.
- Errores.
- Tareas.
- Pruebas de Concepto.

Algunas de sus características más relevantes son:

- Sus elementos están ordenados según la prioridad.
- Su contenido está directamente relacionado con lo acordado con el patrocinador del proyecto.
- Cada elemento que hace parte del Product Backlog se llama PBI (Product Backlog Ítem).

7.1.3.1 - Clasificando el Product Backlog

Durante la construcción del Product Backlog es importante clasificar los elementos del Product Backlog, para lo cual se podría utilizar el concepto de "épicas".

Las épicas permiten agrupar los elementos del Product Backlog, permitiendo una mejor navegación por el mismo. Algunas ideas de las que se podrían definir las épicas son:

- Módulos.
- Componentes.
- Hitos.
- Entregables.
- Funcionalidades.

7.1.4 - Priorizar el Backlog (4)

Priorizar los elementos del Product Backlog es una práctica clave para garantizar la entrega de valor al cliente. Algunos de los factores que se deben considerar para la priorización de las Historias de usuario dentro del Product Backlog son:

- Tiempo.
- Esfuerzo.
- Valor Funcional.
- Dependencias.
- Riesgos.
- Opinión de los Stakeholders.
- Opinión del equipo de desarrollo.
- Valor de Mercado (\$).

Nota: La prioridad de los elementos puede cambiar durante la ejecución del proyecto.

7.1.4.1 – Priorización por urgencia

Una de las técnicas de priorización del Product Backlog está determinada por la urgencia y el valor que representan los elementos del Product Backlog para las Partes Interesadas.

Los elementos que aporten más valor al negocio y tengan una mayor urgencia tendrán una mayor prioridad dentro del Product Backlog:

Valor al Negocio	5	5	10	15	20	25
	4	4	8	12	16	20
	3	3	6	9	12	15
	2	2	4	6	8	10
	1	1	2	3	4	5
		1	2	3	4	5
		Urgencia				

Ilustración 30 - Priorización por urgencia

7.1.4.2 - Visual Story Mapping

Esta es una técnica para proporcionar un esquema visual del producto y sus componentes claves. El Visual Story Mapping, fue formulado por Jeff Patton en 2005, y es comúnmente utilizado para ilustrar la trayectoria del producto.

Este representa la secuencia de iteraciones de desarrollo de los productos (de izquierda a derecha se organizan los componentes según su prioridad de desarrollo, tal como se ve en la imagen):

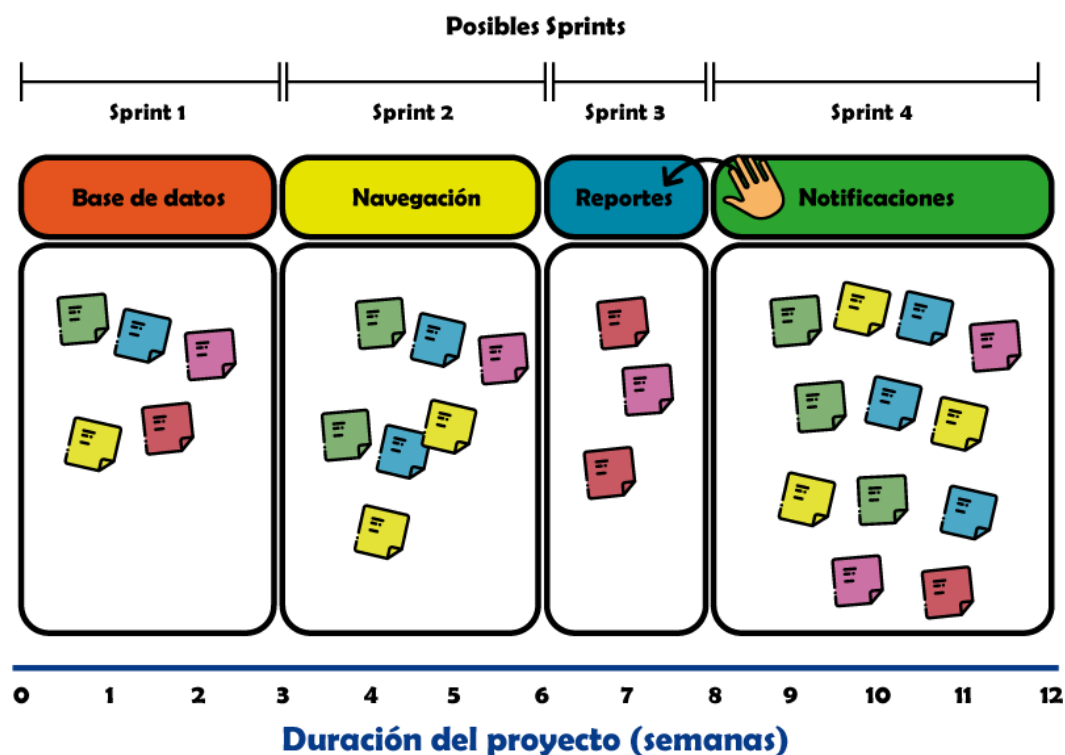


Ilustración 31 - Planificación por componentes o Visual Story Mapping

7.1.5 - Definir el cronograma de entregas (5)

En esta práctica, el Product Owner en compañía de los demás miembros del Equipo Scrum, definen el cronograma de alto nivel del proyecto.

En esta práctica, podrían determinarse la duración estimada de los diferentes Sprints (dado que se ya tienen los elementos del Product Backlog priorizados).

Este cronograma incluye la descripción de alto nivel de las actividades, compromisos, tareas asignados específicamente a los miembros del equipo de desarrollo, los cuales se irán refinando a lo largo del proyecto, en las reuniones de Planificación de cada sprint.

Calendario de Proyecto

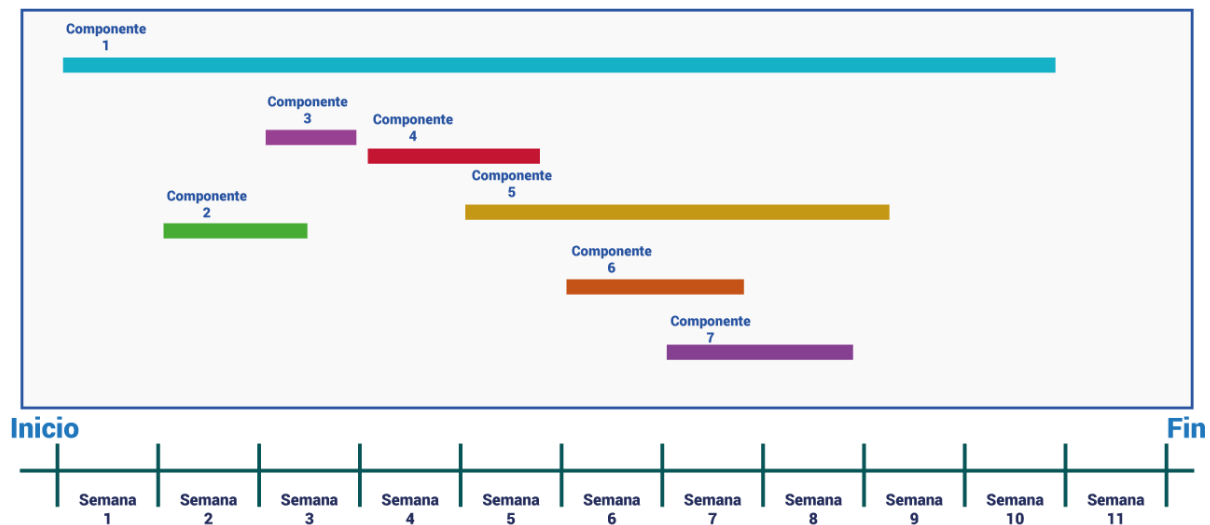


Ilustración 32 - Calendario de alto nivel en un proyecto

Nota: Este cronograma se irá actualizando de manera iterativa durante los Sprints.

7.1.5.1 - Calendario del equipo

El calendario del equipo contiene información sobre la disponibilidad de los miembros del equipo, considerando la información correspondiente a:

- Las vacaciones de los miembros del Equipo Scrum.
- Fechas de licencia.
- Acontecimientos importantes en la organización.
- Días festivos.
- Eventos de la oficina.
- Proyectos en curso.
- Etc.

7.1.6 - Definir la Arquitectura de Producto (6)

El objetivo de esta práctica es establecer o refinar el diseño técnico del producto o del componente de producto a desarrollar. Aunque esta actividad es obligatoria siempre al inicio del proyecto, podría hacerse también de forma iterativa antes de desarrollar componentes en los distintos Sprint.

Para garantizar que la arquitectura de producto que se defina cumple exactamente con los requerimientos del cliente se debe hacer en conjunto entre el equipo de desarrollo y el Product Owner.

Antes de iniciar la construcción del producto es importante identificar la relación entre todos los componentes que harán parte del mismo.

Ejemplo: Una empresa de desarrollo de software planea construir una aplicación para realizar cursos virtuales. Al ser un proyecto de software, en la definición de arquitectura se deberán tener en cuenta los siguientes elementos:

- Bases de datos.
- Servidores de almacenamiento.
- Servidores de procesamiento.
- Nombres de dominio.
- Servidores de procesamiento de correo electrónico.
- Firewall.
- Web services.
- Entre otros.

7.1.6.1 - Sprint Cero

Esta iteración es la primera en realizarse. El objetivo del Sprint Cero es preparar el equipo de proyecto desde una perspectiva tecnológica, metodológica y organizativa, buscando conformar un equipo y no simplemente un grupo de personas.

En esta actividad se analizan y evalúan las posibles soluciones que se pueden establecer para el desarrollo del proyecto.

Estas posibles soluciones pueden ser:

- Herramientas.
- Soluciones Tecnológicas.
- Soluciones técnicas.

Por cada solución escogida se deberá realizar una prueba de concepto, en donde se deben tener en cuenta los siguientes criterios:

- Curva de aprendizaje.
- Escalabilidad (Solo si es solución técnica o tecnológica).
- Soporte.
- Costo.

Nota: Este sprint solo se utiliza para los casos en que el equipo de proyecto tiene poca o nula experiencia en la tecnología en la cual se va a construir el producto o incluso si cuenta con poca o nula experiencia en el tipo de producto que se va a desarrollar.

7.1.6.2 - Evaluación y Selección de Proveedores

El propósito de esta actividad es garantizar la selección objetiva de los proveedores, utilizando los criterios definidos por la organización. (Ver más información en la sección - Contratación de Proveedores).

7.2 - Etapa 2: Planificación

ID	Práctica	Rol vs Nivel de Involucramiento
Etapla 2: Planificación		
7	7.2.1 - Escribir las historias de usuario y tareas (7)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Medio • Equipo de desarrollo: Alto
8	7.2.2 - Priorizar las Historias de Usuario (8)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Medio • Equipo de desarrollo: Alto

9	7.2.3 - Reunión de Planificación del Sprint (9)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Alto • Equipo de desarrollo: Alto
---	---	---

7.2.1 - Escribir las historias de usuario y tareas (7)

7.2.1.1 - ¿Qué son las Historias de Usuario?

Las historias de usuario son una forma más entendible y precisa para la escritura de los requerimientos a desarrollar en el proyecto.

Están compuestas por 3 elementos principalmente:

- **Como:** Describe el Rol de la persona o grupo que solicita (o usaría) la funcionalidad o requerimiento.
- **Quiero:** Describe la necesidad o requerimiento del usuario, por lo general, es una frase corta.
- **Para:** Describe el beneficio esperado por el usuario una vez se desarrolle el requerimiento.

Bill Wake inventó el acrónimo INVEST para describir las características de una buena historia de usuario:

- **Independiente:** Las historias pueden completarse en cualquier orden.
- **Negociable:** Los detalles de la historia son co-creados por los desarrolladores y los clientes durante el desarrollo.
- **Valiosa:** La funcionalidad es valiosa para los clientes o los usuarios del producto.
- **Estimable:** Los desarrolladores pueden encontrar una estimación razonable para construir la historia.
- **Pequeña (Small):** Las historias deberían construirse en poco tiempo, generalmente alrededor de "días/persona". Se tiene que poder construir muchas historias en una iteración.
- **Probable (Testeable):** Se debe poder escribir pruebas que verifiquen que el producto de la historia funcione adecuadamente.

7.2.1.2 - Mockups y prototipos iniciales de proyecto

Los Mockups representan el diseño del producto antes de su desarrollo, consideran el flujo que debe seguir el producto, y sirven para mostrar las posibles funcionalidades al cliente y así confirmar que estas entregarán el valor esperado.

Los mockups normalmente son un conjunto de bocetos, diseños, diagramas y/o representaciones; y es especialmente necesario contar con un mockup o prototipo previo al Sprint, pues éstos servirán de guía a los miembros del equipo.

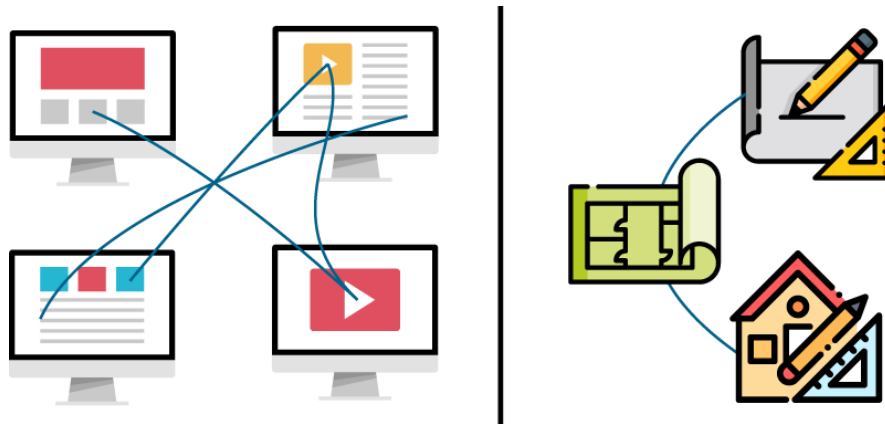


Ilustración 33 – Mockups

7.2.1.3 - Planificación en paralelo

Una actividad que puede representar una gran diferencia es la planificación en paralelo. Esta actividad consiste en formar un pequeño equipo de personas que se encargan de realizar las actividades de planificación paralelamente al desarrollo de los elementos realizados. Con esto se logrará que las reuniones de planificación del Sprint sean muchísimo más eficientes. Algunas de las cosas que se puedan hacer en paralelo al desarrollo son:

- Realizar las entrevistas o investigaciones necesarias para escribir las historias de usuario.
- Realizar los prototipos y confirmar el valor con el usuario.

7.2.1.4 - Desglosar las Historias de Usuario en tareas

Esta actividad sirve para identificar las posibles dependencias entre las historias de usuario, para ello, se elabora un listado con todas las tareas que se deben llevar a cabo para completar la Historia de Usuario.

Normalmente se distinguen 4 tipos de dependencias entre historias de usuario:

- Dependencias obligatorias.
- Dependencias discrecionales.
- Dependencias externas.
- Dependencias internas.

Algunas técnicas que pueden ser usadas para desglosar las Historias de Usuario en tareas son:

- Los Diagramas de Flujo.
- Los Diagramas de Gantt.

7.2.2 - Priorizar las Historias de Usuario (8)

Esta práctica está relacionada con la planificación en paralelo, debido a que mientras el equipo termina de desarrollar lo establecido para el último Sprint, el Product Owner realiza una priorización de las historias de usuario que pueden considerarse para el siguiente Sprint y que se pueden tomar como base para guiar el rumbo del equipo.

Es importante que esta priorización se realice antes de que se lleve a cabo la Reunión de Planificación del Sprint con el fin de optimizar el tiempo.

Los elementos que se deben considerar para la priorización de las historias de usuario son:

- Valor para el negocio.
- Riesgo.
- Dependencia.

7.2.3 - Reunión de Planificación del Sprint (9)

7.2.3.1 - Seleccionar el trabajo a desarrollar

El Product Owner expone el objetivo que el Sprint debería lograr y los elementos del Product Backlog que, si se completan en el Sprint, lograrían el objetivo del Sprint, con esto el Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint.

La entrada a esta reunión está constituida por el Product Backlog y tomando la priorización de historias de usuario realizada por el Product Owner, incluyendo además:

- El último incremento de producto.
- La capacidad proyectada del Equipo de Desarrollo para el Sprint.
- La velocidad del Equipo de Desarrollo.
- Fallos encontrados al producto o incrementos de producto (registro de errores).

El número de elementos del Product Backlog seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el Sprint que comienza.

Durante la Planificación del Sprint el Equipo Scrum define el objetivo del Sprint. El objetivo del Sprint debería lograrse durante el Sprint a través de la implementación del Product Backlog y proporciona una guía al Equipo de Desarrollo del por qué se está construyendo el incremento.

7.2.3.2 - Estimación del trabajo seleccionado

Es muy importante realizar la estimación del trabajo seleccionado (comúnmente escrito en forma de historias de usuario) para poder hacer planificaciones más precisas.

Para garantizar una estimación más precisa, se deberían considerar criterios como:

- Tamaño.
- Complejidad.
- Duración.
- Cantidad de recursos.

- Riesgos.
- Limitaciones.

Normalmente las técnicas de estimación usadas por Scrum están basadas en el juicio de expertos, sin embargo existen otras técnicas que se pueden utilizar y combinar:

Planning Poker

El póker de planificación consiste en un conjunto de tarjetas numéricas que sirven para realizar la estimación de tareas o historias de usuario.

- El póker puede ir numerado de 1 a 10 o utilizando la sucesión de Fibonacci.
- El uso del póker promueve una mayor interacción y una mejor comunicación entre los participantes, además de hacer más dinámica la reunión de planificación del Sprint.

Fist of Five

Es una técnica que permite lograr el consenso en los Equipos Scrum. Se hace usando los dedos de la mano, donde el número de dedos que se utiliza para la votación indica el nivel de acuerdo y el deseo para el debate:

- **Un dedo:** no estoy de acuerdo con la conclusión del grupo y tienen grandes preocupaciones.
- **Dos dedos:** no estoy de acuerdo con la conclusión del grupo y me gustaría hablar de algunos problemas menores.
- **Tres dedos:** no estoy seguro y me gustaría asumir la conclusión de consenso del grupo.
- **Cuatro dedos:** Estoy de acuerdo con la conclusión del grupo, pero me gustaría discutir algunos problemas menores.
- **Cinco dedos:** Estoy totalmente de acuerdo con la conclusión del grupo.

7.2.3.3 - Compromiso del equipo

Una de las actividades clave durante la Planificación de un Sprint es la apropiación de las Historias de Usuario y tareas por parte del equipo de desarrollo, esto se logra cuando los miembros del equipo se auto-assignan el trabajo a desarrollar, a su vez que adquieren el compromiso público de terminar dicho trabajo.

Es sumamente importante garantizar que existe equilibrio en la cantidad de trabajo seleccionado por cada uno de los integrantes del equipo.

Suele suceder que en equipos Scrum donde se encuentran combinados miembros expertos y miembros novatos existirá un “desequilibrio” en etapas tempranas del proyecto, pues los miembros expertos contarán con más capacidad de trabajo mientras los miembros novatos se nivelan en su curva de aprendizaje.

7.2.3.4 - ¿Cómo se conseguirá “terminar” el trabajo seleccionado?

Una vez que se ha establecido el objetivo y se han seleccionado los elementos del Product Backlog para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado” durante el Sprint. Los elementos del Product Backlog seleccionados para este Sprint, junto con el plan para terminarlos, recibe el nombre de Sprint Backlog.

Durante la Planificación del Sprint se debe asegurar que se planifica el suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en el Sprint que comienza.

Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del Sprint es descompuesto en unidades de un día o menos. El Equipo de Desarrollo se autoorganiza para asumir el trabajo del Sprint Backlog, tanto durante la Planificación del Sprint como a lo largo del Sprint.

El Product Owner puede ayudar a clarificar los elementos del Product Backlog seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos del Product Backlog seleccionados por el Product Owner. El Equipo de Desarrollo podría también invitar a otras personas a que asistan para proporcionar asesoría técnica o relacionada con el dominio.

7.2.3.4.1 - Objetivo del Sprint

El objetivo del Sprint es una meta que se establece para cada Sprint durante la Reunión de Planificación del Sprint. Dicha meta describe el alcance de cada Sprint en alineación con el Product Backlog, a su vez proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento.

Definir el objetivo del Sprint brinda al Equipo de Desarrollo cierta flexibilidad con respecto a la funcionalidad que ha de ser construida en el Sprint.

- Los elementos seleccionados del Product Backlog ofrecen una orientación sobre lo que puede ser el objetivo del Sprint.
- El objetivo del Sprint garantiza que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.
- Por lo general el objetivo del Sprint suele ser una frase corta que explica de forma simple el incremento esperado al final del Sprint.

A medida que el Equipo de Desarrollo trabaja mantiene el objetivo del Sprint en mente. Si el trabajo resulta ser diferente de lo esperado, los miembros del equipo escalan la situación con el Scrum Master para contactar al Product Owner y se pueda negociar el alcance del Sprint.

7.2.3.4.2 - Spikes

Un Spike sirve para incluir en un sprint tareas que NO implican desarrollar una historia de usuario y por tanto NO aportan directamente al incremento de producto que se está desarrollando.

Algunos ejemplos de Spikes son:

- Capacitación del Equipo.
- Documentación del proyecto y/o producto (ejemplo: Documentar el código fuente).
- Despliegues / Implementaciones.
- Adopción de nuevas herramientas.

7.3 - Etapa 3: Desarrollo del Sprint

ID	Práctica	Rol vs Nivel de Involucramiento
Etapa 3: Desarrollo del Sprint		

10	7.3.1 - Desarrollar los entregables (10)	<ul style="list-style-type: none"> • Product Owner: Bajo • Scrum Master: Medio • Equipo de desarrollo: Alto
11	7.3.2 - Scrum diario (11)	<ul style="list-style-type: none"> • Product Owner: Nulo • Scrum Master: Alto • Equipo de desarrollo: Alto

7.3.1 - Desarrollar los entregables (10)

El objetivo de esta actividad es el desarrollo de los entregables del Sprint, los manuales o documentación relacionada considerando siempre la definición de “terminado”.

Nota: En conjunto con esta práctica, se ejecutan los Scrums Diarios, y Las pruebas de los entregables.

7.3.1.1 – El ciclo de desarrollo para proyectos de software

Para el desarrollo de proyectos de software, normalmente se sigue el flujo que se muestra en el gráfico a continuación:

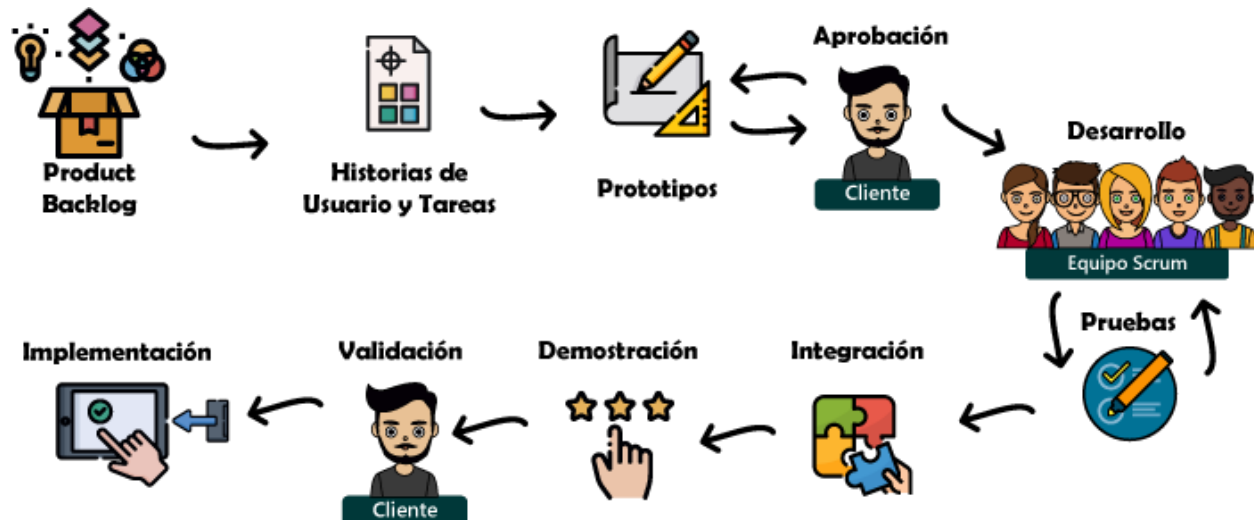


Ilustración 34 - Ciclo de vida de desarrollo de proyectos de Software

7.3.1.1.1 – Las pruebas

Las Pruebas son una forma de comprobar el correcto funcionamiento del producto. Por lo general existen varios tipos de pruebas, siendo los principales:

- **Pruebas unitarias:** Son escritas por el mismo desarrollador siguiendo los criterios de aceptación de la historia de usuario. Estas pruebas suelen realizarse de manera automatizada.
- **Pruebas de colegas:** Suelen ser manuales, en las cuales el colega revisan no solo el funcionamiento, sino la calidad del código, y compara el resultado vs prototipos.
- **Pruebas de integración:** Suelen ser automatizadas. Éstas se ejecutan en cada integración y antes de cada entrega.

También se distinguen 2 tipos de ejecución de las pruebas:

- Pruebas manuales.
- Pruebas automatizadas.

Las 2 posibles salidas de una prueba son:

- **Aprobada:** Se cumple con todas las expectativas y/o criterios de aceptación de las historias de usuario que se están probando.
- **Rechazada:** NO se cumple con todas las expectativas y/o criterios de aceptación de las historias de usuario que se están probando. Cuando se rechaza una historia de usuario durante las pruebas, se debe reportar el error en un "Log de Errores", normalmente dentro del Product Backlog.

Reporte de errores

Es importante tener en cuenta que los errores deben pasar por un proceso de estimación igual que las Historias de Usuario (se planifica su resolución en los diferentes Sprints del proyecto).

Cuando se reporten los errores, se debe tener en cuenta:

- Evitar escribir declaraciones vagas o suposiciones no probadas.
- Poner títulos genéricos a los errores.
- Se deberían categorizar los errores, ej: Errores en producción, Errores de colegas, Errores en la revisión, etc.
- Deberían describirse como pasos de reproducción sencillos y repetibles para que la persona que corregirá el error pueda seguir la secuencia.

7.3.1.1.2 – Integración continua

La Integración Continua es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas para asegurar la integridad del código.

Para facilitar la integración continua:

- Se debe tener un repositorio único para el código fuente.
- Se debe garantizar que el código se actualiza constantemente tanto en el repositorio, como en el ambiente de trabajo de cada desarrollador.
- Siempre ejecutar pruebas automatizadas para garantizar que cuando se mezcla el código, no se pierden o dañan las funcionalidades existentes.

7.3.1.1.3 – Documentación

La documentación es un aspecto vital en los proyectos de software, esto permitirá el posterior entendimiento del código fuente y así garantizar la propiedad colectiva del producto. Para que la documentación sea efectiva, se debe garantizar que:

- Sea fácil de redactar y entender.
- Debe estar compartida con todos.
- Se actualiza diariamente.
- Debe tener un índice o búsqueda.
- Debe estar organizada por módulos o componentes.
- Debe ser alimentada por todo el equipo.

7.3.1.1.4 – Refactorización

El objetivo de esta técnica es mejorar el mantenimiento del código existente y hacerlo más simple, más conciso y más flexible. Refactorizar significa mejorar el diseño del código actual, sin cambiar el comportamiento del código. Algunas de las ventajas de la refactorización son:

- Mejorar la facilidad de comprensión del código.
- Mejorar su estructura y diseño.
- Eliminar código muerto.
- Facilitar el mantenimiento en el futuro.

Algunos de los momentos clave para realizar la refactorización, son:

- Después de pasar una prueba automatizada.
- Cuando agregar una nueva característica es bastante complicado.
- Cuando mejora el conocimiento o experiencia del equipo de desarrollo.
- Cuando el equipo dedica bastante tiempo a entender el detalle del código.

Algunos ejemplos de refactorización son:

- Organizar el repositorio de documentos/archivos.
- Separación de funciones.
- Renombrar variables.
- Simplificación de las interfaces.

7.3.2 - Scrum diario (11)

Para más detalles ver la *Sección 3.3 - Daily Scrum (Scrum Diario)*

7.3.2.1 - Seguimiento del Progreso del Sprint

En cualquier momento durante un Sprint es posible conocer el progreso del Sprint sumando el trabajo restante total en los elementos del Sprint Backlog. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el objetivo del

Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint el Equipo de Desarrollo puede gestionar su progreso.

7.4 - Etapa 4: Revisión del Sprint

ID	Práctica	Rol vs Nivel de Involucramiento
Etapa 4: Revisión del Sprint		
12	7.4.1 - Reunión de Revisión del Sprint (12)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Alto • Equipo de desarrollo: Alto
13	7.4.2 - Reunión de Retrospectiva del Sprint (13)	<ul style="list-style-type: none"> • Product Owner: Nulo • Scrum Master: Alto • Equipo de desarrollo: Alto

7.4.1 - Reunión de Revisión del Sprint (12)

En esta reunión de revisión se realiza la demostración del incremento de producto al Product Owner y partes interesadas invitadas, con el fin de obtener aprobación sobre lo que se esperaba y lo que fue desarrollado por el equipo.

A continuación, se listan las actividades que se realizan durante la Reunión de Revisión del Sprint:

- **Presentar el incremento de producto:** El equipo de desarrollo presenta al Product Owner y al Scrum Master el incremento del producto desarrollado.
- **Probar el incremento de producto (Validación):** El Product Owner y las partes interesadas invitadas (solo si las hay) prueban el incremento de producto con base en la comparación del desarrollo frente a cada historia de usuario y sus criterios de aceptación. (Es válido que se cuente con una lista de chequeo para llevar el registro de lo que se cumple y lo que no se cumple).
- **Aprobación del incremento de producto:** Con base en el incremento presentado y la validación el Product Owner da la aprobación del incremento de producto. En caso que el incremento no cumpla con lo esperado puede darse el Rechazo del incremento.
- **Monitoreo, control y seguimiento del proyecto:** Ver más en *Sección 6.2 - Seguimiento y control del proyecto*.

- **Ritmo/velocidad del equipo:** Se identifica el ritmo/velocidad del Equipo de Desarrollo.
- **Compromisos del equipo:** Se identifican compromisos por parte del equipo en caso del rechazo del incremento de producto o de historias de usuario puntuales.
 - En caso de rechazo se agregan las mejoras al Product Backlog con la prioridad más alta posible, con el fin de corregirlas en el próximo sprint de ser posible.
- **Actualizar el Product Backlog:** Se debe actualizar el Product Backlog y dar por terminado el sprint actual.

¿Cómo se realizan las validaciones?

En esta reunión se presenta el incremento de producto ante los interesados, los cuales se cerciorarán que cada una de las funcionalidades establecidas y solicitadas en el Sprint estén terminadas (bajo los lineamientos de los criterios de aceptación y los criterios de “terminado”).

7.4.1.1 - Deuda técnica

La deuda técnica se refiere al trabajo que los equipos omiten o no se completa durante uno o varios Sprints.

- Si la deuda técnica se acumula, puede conllevar a un mantenimiento, integración y costos elevados en el despliegue del producto.
- La priorización frecuente de Historias de Usuario contribuye a disminuir la deuda técnica.
- Cualquier deuda técnica no debería llevarse más allá de un sprint.

Algunas de las causas de la deuda técnica son:

- Evaluación inadecuada o incompleta de Historias de Usuario
- Falta de coordinación entre los miembros del equipo
- Intercambio deficiente del conocimiento empresarial

7.4.1.2 - Refinamiento del Product Backlog

El refinamiento del Product Backlog es el acto de añadir detalle, estimaciones y orden a los elementos del Product Backlog; se trata de una actividad continua en la cual el Product Owner y el Equipo de Desarrollo examinan, revisan y detallan los elementos del Product Backlog.

- Los elementos del Product Backlog pueden actualizarse en cualquier momento por El Product Owner o a criterio suyo.
- El refinamiento usualmente consume no más del 10% del tiempo de la Reunión de Revisión del Sprint.

7.4.2 - Reunión de Retrospectiva del Sprint (13)

La Retrospectiva del Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de crear un plan de mejoras que sean abordadas durante el siguiente Sprint. La Retrospectiva debería realizarse en un ambiente liberador que permita al equipo fluir todo tipo de ideas.

La Retrospectiva del Sprint tiene lugar después de la Revisión de Sprint (Sprint Review) y antes de la siguiente Planificación de Sprint. Se trata de una reunión restringida a lo **máximo de 4 horas para Sprints de un mes**.

El Scrum Master enseña a todos a mantener el evento positivo y productivo; y participa de esta reunión como un miembro del equipo ya que parte de la responsabilidad de los incrementos recae sobre él.

El Scrum Master motiva al equipo para que mejore su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint. Durante cada Retrospectiva del Sprint el Equipo Scrum identifica y planifica formas de mejorar la calidad del producto mediante el mejoramiento de la calidad de las prácticas, que pueden ser implementadas en el próximo Sprint.

El hecho de implementar estas mejoras en el siguiente Sprint constituye la adaptación subsecuente a la inspección del Equipo de Desarrollo mismo. Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva del Sprint ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

En resumen, durante la Retrospectiva:

- El equipo de trabajo se enfoca en responder a las siguientes preguntas:
 - Qué cosas han funcionado bien.
 - Cuáles hay que mejorar.

- Qué cosas quiere probar el equipo en la siguiente iteración.
- Qué ha aprendido el equipo.
- El equipo identifica las mejoras accionables para mantener o mejorar el trabajo del equipo, y mantener la motivación de los miembros.
- El equipo documenta y actualiza el registro de lecciones aprendidas.

7.5 - Etapa 5: Implementación

ID	Práctica	Rol vs Nivel de Involucramiento
Etapa 5: Implementación		
14	7.5.1 - Planificación de la implementación (14)	<ul style="list-style-type: none"> • Product Owner: Medio • Scrum Master: Bajo • Equipo de desarrollo: Alto
15	7.5.2 - Implementación de entregables (15)	<ul style="list-style-type: none"> • Product Owner: Medio • Scrum Master: Bajo • Equipo de desarrollo: Alto

El objetivo de esta etapa es la puesta en producción de los incrementos aprobados en la Revisión del Sprint, para su inmediata utilización y aprovechamiento por parte del cliente y/o interesados. En los proyectos Scrum, suele ser una etapa iterativa, para así aprovechar todos los incrementos utilizables desde etapas tempranas del proyecto.

Según la naturaleza de la organización, es común que éstas prácticas sean ejecutadas por grupos distintos al Equipo Scrum, sin embargo es altamente recomendable que sea el mismo Equipo el que se haga responsable de este trabajo, así se garantiza un real compromiso por la calidad de los entregables, a la vez que se evita la aparición de una cultura de la culpa.

7.5.1 - Planificación de la implementación (14)

El objetivo de esta práctica es preparar todos los elementos necesarios para realizar una implementación exitosa de los incrementos generados en los Sprint, disminuyendo significativamente los riesgos que puedan impactar negativamente las operaciones de los usuarios/clientes.

La cantidad de entregas (despliegues) dependerá de lo acordado con el cliente o patrocinador del proyecto, y por lo general, dependen del valor para la organización.

7.5.1.1 - Coordinación con operaciones

Una actividad clave es la coordinación con los distintos equipos de operaciones del cliente para preparar la implementación.

Algunos elementos clave en esta actividad son:

- ¿Cuándo se hará la implementación?
- ¿Qué usuarios recibirán el nuevo producto?
- ¿La implementación del producto afectará las operaciones?
- ¿Cómo se notificará a los usuarios?
- ¿Los usuarios están capacitados para usar el nuevo producto?
- ¿Qué hará el Equipo de Desarrollo en caso de que la implementación tenga fallos?

Nota: Es altamente recomendable que los equipos de operaciones hayan participado de las reuniones de Revisión de los Sprint.

7.5.2 - Implementación de entregables (15)

Esta práctica busca poner a disposición de los usuarios los incrementos finalizados y utilizables previamente desarrollados por el Equipo Scrum.

Cabe aclarar que esta práctica no es posible aplicarla en todos los tipos de proyectos, ni es obligatorio ejecutarla al finalizar cada Sprint, solo será aplicable cuando el incremento de producto genere valor para los usuarios o para la organización.

La implementación de entregables puedes realizarse de varias formas, las 2 más comunes son:

- **Big bang:** Este tipo de implementación, busca poner el incremento a disposición de toda la comunidad de usuarios al mismo tiempo. También se le llama “Implementación masiva”.
- **Por fases:** Este tipo de implementación se usa en caso de que se quiera hacer una segmentación de los usuarios para implementar el incremento (ej: entregarlos solamente a usuarios específicos).

7.5.2.1 - Confirmación de implementación exitosa

Una vez el incremento de producto está en ambientes productivos, es importante confirmar que no hubo afectaciones a la operación, por lo que podrían ejecutarse revisiones o pruebas post-implementación.

En muchas ocasiones luego de la confirmación de implementación exitosa, se realiza la notificación formal al cliente (incluso se firma algún artefacto como constancia), y así disminuir la probabilidad de que surjan nuevas solicitudes de cambios sobre los elementos ya entregados.

7.5.2.2 - Despliegues Fallidos

En caso de que los resultados de la implementación sean negativos se evaluará de manera rápida si se pueden solucionar durante una ventana de mantenimiento, en caso contrario se deberá realizar un “rollback” y planear un nuevo Sprint en el que se solucionen los problemas encontrados, dando paso a un nuevo despliegue.

7.6- Etapa 6: Cierre del proyecto

ID	Práctica	Rol vs Nivel de Involucramiento
Etapla 6: Cierre del proyecto		
16	7.6.1 - Cierre del proyecto (16)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Bajo • Equipo de desarrollo: Bajo

17	7.6.2 - Reunión de Retrospectiva del proyecto (17)	<ul style="list-style-type: none"> • Product Owner: Alto • Scrum Master: Alto • Equipo de desarrollo: Medio
----	--	--

7.6.1 - Cierre del proyecto (16)

En la reunión de cierre de proyecto se dejará registro del cierre y motivo del cierre del proyecto, para este fin podría construirse un acta de Cierre del proyecto.

Los siguientes pueden ser motivos para el cierre del proyecto:

- **Cierre exitoso del proyecto:** Se cumplió con todos los entregables programados dentro del presupuesto asignado y de acuerdo con los criterios de aceptación (satisfacción total del cliente).
- **Cierre parcial del proyecto o aplazamiento:** Se cumplió parcialmente con los entregables programados, se está superando el presupuesto asignado o no se ha cumplido completamente con todos los criterios de aceptación (satisfacción parcial del cliente).
- **Cancelación del proyecto:** No se cumplió con ninguno de los entregables programados, se agotó el presupuesto asignado antes de tiempo, no se cumplió con ningún criterio de aceptación (insatisfacción total del cliente).

¿Qué hacer durante esta etapa?

- Entregar el producto terminado.
- Obtener la aprobación de los entregables del proyecto por parte del cliente.
- Cerrar formalmente el proyecto.
- Verificar la satisfacción por parte del cliente y las partes interesadas.

¿Qué se necesita para la reunión?

- **Producto o incremento de producto desarrollado por el equipo:** Para registrar el estado del producto al momento de cierre del proyecto.
- **Acta de Inicio del Proyecto:** Para validar el alcance, objetivos y aspectos iniciales del proyecto.

- **Resultados de las pruebas de integración:** Incluye un informe de las pruebas que se realizaron tras la entrega de cada incremento del producto para garantizar la correcta integración entre componentes.
- **Criterios de aceptación del producto:** Para verificación y validación de lo esperado en comparación con lo que se entregó.
- **Documentación/Capacitación al cliente:** Para garantizar que el cliente cuenta con el conocimiento sobre el uso, instalación, configuración o soporte básico del producto.
- **Informes de proyecto:** Que muestran el estado actual del progreso del proyecto (Se mostrarán informes como: diagrama de flujo acumulado y diagrama de presupuesto).

Es importante que quede un artefacto como evidencia del cierre del proyecto, con lo cual se garantiza la terminación oficial de las actividades del proyecto, o el cumplimiento de los compromisos posteriores (en caso que existan).

7.6.2 - Reunión de Retrospectiva del proyecto (17)

Dentro de los objetivos de esta práctica encontramos:

- Analizar y aprender de los aciertos y errores que se cometieron a lo largo de todo el proyecto.
- Analizar los informes generados por Aseguramiento de Calidad durante todo el proyecto.
- Crear un Informe de mejoras y lecciones aprendidas a lo largo del proyecto.

¿Quiénes participan de la reunión?

- Scrum Master.
- Product Owner.
- Equipo de Desarrollo.
- En algunas ocasiones incluso podría participar el cliente.

8 Scrum en Grandes Proyectos

Scrum está diseñado para funcionar en proyectos de cualquier tamaño, por lo que, haciendo algunos ajustes a lo explicado en las anteriores secciones, puede ser utilizado en grandes proyectos.

8.1 – Distribución de equipos

Cuando un Proyecto es muy grande y un solo Equipo Scrum no es suficiente para desarrollar todo el trabajo, será necesario contar con múltiples equipos. Algunas de las consideraciones que se deben tener en cuenta para realizar la distribución de los equipos son:

- Hasta donde sea posible, se debería contar con un solo Product Owner para el proyecto.
- El tamaño de los equipos debería mantenerse equilibrado.
- No deberían conformarse equipos especializados, los equipos multifuncionales avanzan más rápido y generan menos dependencias.
- No deberían cambiarse los miembros de equipo durante la ejecución del proyecto.

8.1.1 – Program Owner

Cuando los proyectos hacen parte de un programa, el rol de Program Owner toma bastante relevancia. El Program Owner será el rol responsable de:

- Gestionar el Product Backlog del programa.
- Garantizar la entrega de beneficios.
- Aprobar o rechazar los cambios a nivel de programa.
- Coordinar el trabajo con los distintos Product Owner de cada proyecto perteneciente al programa.

Algunos roles que también podrían participar en grandes proyectos son:

- **Agile Coach:** Guía la transformación ágil en las organizaciones.
- **Line Manager:** Conoce la línea de negocio a la que pertenece el programa o el proyecto.

8.2 - Product Backlog en múltiples equipos

A menudo, varios Equipos Scrum trabajan juntos para desarrollar el mismo producto. Para describir el trabajo a realizar sobre el producto se utiliza un único Product Backlog.

- Podría usarse un atributo del Product Backlog para agrupar los elementos que pertenecen a los distintos equipos.

8.3 - Scrum de Scrums

El Scrum de Scrums es el evento enfocado en la coordinación de los Sprints para múltiples equipos dentro del mismo proyecto.

- Este evento por lo general tiene una duración de 15 minutos.
- En algunos proyectos donde la necesidad de comunicación entre equipos es vital (ej: cuando existen dependencias), se realiza diariamente.
- Se debe realizar por lo menos 1 vez por Sprint.
- Es dirigida por un Scrum Master.
- Participan los representantes de cada equipo perteneciente al proyecto.

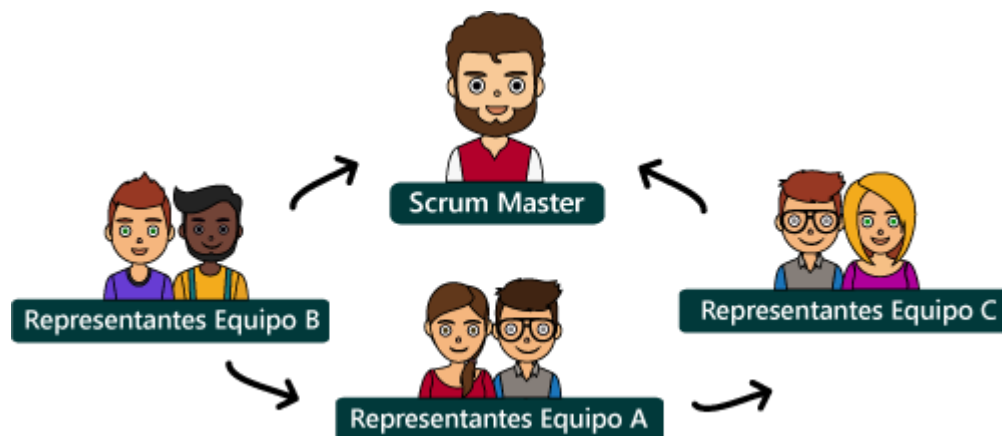


Ilustración 35 - Scrum of Scrums

Objetivos de la reunión

- Coordinar el trabajo de múltiples equipos que trabajan para un mismo proyecto.
- Evitar atrasos por dependencias o riesgos no identificados a tiempo.

Para garantizar efectividad en este evento, se realizan las siguientes preguntas:

- ¿En qué ha estado trabajando tu equipo?
- ¿En qué va a trabajar tu equipo?
- ¿Han tenido impedimentos?
- ¿Existen dependencias entre tu equipo y otros equipos?

8.4 - Coordinación de Sprints

Cuando en un proyecto se cuenta con múltiples equipos, es importante garantizar que sus Sprints se mantienen coordinados en el tiempo, es decir, todos los equipos inician y terminan sus Sprints en las mismas fechas.

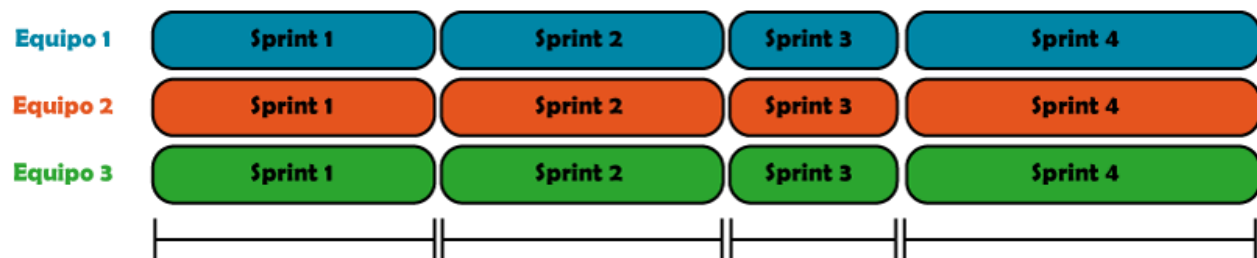


Ilustración 36 - Coordinación de Sprints

Algunas consideraciones a tener en cuenta son:

- Debería realizarse una sola reunión de planificación de Sprint.
- Podría hacerse una sola reunión de revisión del Sprint.
- Cada equipo debe realizar su propia retrospectiva. Sin embargo, es importante hacer una sesión periódica de lecciones aprendidas de todos los equipos para compartir el conocimiento adquirido en cada equipo.

8.5 – Guías de ajuste sobre las prácticas Scrum

Cuando las prácticas y herramientas de Scrum han de ser adoptadas en una organización, estas no siempre se ajustan 100% al contexto del proyecto, el tipo de organización o incluso las necesidades concretas de un cliente específico, razón por la cual en distintas organizaciones la APMO diseña un conjunto de guías que permiten adaptar las prácticas de Scrum al contexto de cada proyecto, estas guías reciben el nombre de “Guías de ajuste”.

- Las guías de ajuste son útiles para los miembros de los distintos equipos Scrum, principalmente para los distintos Product Owner.
- Las guías de ajuste también pueden ser utilizadas para adaptar el programa de auditoría por parte de los grupos de calidad de la organización.

9 Agradecimiento

De los miles de personas que han contribuido a Scrum, distinguimos a quienes aportaron la base para la construcción de este libro: Jeff Sutherland trabajando con Jeff McKenna y John Scumniotales; y Ken Schwaber trabajando con Mike Smith y Chris Martin – basado en “The Scrum Guide”.

Extendemos también el agradecimiento al equipo de trabajo de CertMind quienes durante los últimos meses aportaron su conocimiento y experiencia, haciendo posible el conjunto de mejoras realizadas al Marco de Trabajo, así como la comunidad de expertos que participó en la edición, ajuste y traducción.

10 Términos oficiales en inglés

- **Aseguramiento de la Calidad:** Quality Assurance
- **Daily Scrum:** Scrum Diario
- **Definición de Terminado:** Definition of “Done” (DoD)
- **Diagrama de flujo acumulado:** Cumulative Flow Diagram
- **Equipo de Desarrollo:** Development Team
- **Equipo Scrum:** Scrum Team
- **Incremento:** Increment
- **Objetivo del Sprint:** Sprint Goal
- **Periodos de tiempo limitado:** Time-Boxes
- **Refinamiento:** Refinement
- **Reunión de Planificación del Sprint:** Sprint Planning Meeting
- **Revisión del Sprint:** Revisión del Sprint
- **Retrospectiva del Sprint:** Sprint Retrospective
- **Scrum Diario:** Daily Scrum
- **Solicitud de cambio:** Request For Change
- **Tareas de investigación:** Research Tasks

10. Lista de figuras

Ilustración 1 - Estructura de la guía	11
Ilustración 2 - Metodologías ágiles vs Metodologías Cascada.....	12
Ilustración 3 - El Equipo Scrum.....	13
Ilustración 4 - Product Owner	14
Ilustración 5 - El Equipo de Desarrollo	15
Ilustración 6 - El Scrum Master	16
Ilustración 7 - El Scrum Master guía al equipo.....	18
Ilustración 8 - El Sprint.....	20
Ilustración 9 - Scrum Board.....	28
Ilustración 10 - Burndown chart del Sprint.....	30
Ilustración 11 - Diagrama de flujo acumulado	31
Ilustración 12 - El cliente participa en el proyecto ágil	36
Ilustración 13 - Equilibrio entre burocracia y anarquía: agilidad.....	40
Ilustración 14 - Habilidades del Equipo Scrum	44
Ilustración 15 – Matriz de competencias	45
Ilustración 16 - Necesidades del equipo.....	45
Ilustración 17 - Fórmula para el cálculo del ROI.....	48
Ilustración 18 - Freelancers	50
Ilustración 19 - Actividades de la APMO	51

Ilustración 20 - Velocidad del Equipo Scrum	52
Ilustración 21 - Apetito de riesgo	56
Ilustración 22 - Ciclo de vida de la gestión de riesgos.....	57
Ilustración 23 - Criterios para la evaluación de riesgos	58
Ilustración 24 - Acciones de mitigación de riesgo	60
Ilustración 25 - Flujo de cambios	62
Ilustración 26 - Lean Canvas	64
Ilustración 27 - Ciclo de vida en un proyecto Scrum	65
Ilustración 28 - Restricciones del Proyecto	67
Ilustración 29 - Modelo de desarrollo de equipos – Dr. Bruce Tuckman	70
Ilustración 30 - Priorización por urgencia	75
Ilustración 31 - Planificación por componentes o Visual Story Mapping.....	76
Ilustración 32 - Calendario de alto nivel en un proyecto	77
Ilustración 33 – Mockups	81
Ilustración 34 - Ciclo de vida de desarrollo de proyectos de Software	87
Ilustración 35 - Scrum of Scrums	101
Ilustración 36 - Coordinación de Sprints	102

11. Cambios respecto a la edición 2017.

- **1.1:** Se mejora la explicación sobre la historia de Scrum
- **1.8:** Nueva sección
- **5.2.3.1:** Nueva sección "Gestionar el conocimiento"
- **5.3:** Se mejora la explicación sobre el principio de "Simplicidad"
- **6.2.1:** Nueva Sección "Oficina de Gestión de Proyectos Ágiles (APMO)"
- **6.3:** Se agregaron ejemplos de algunos riesgos posibles
- **7.16:** Nueva práctica
- **7.3.1.1:** Nueva explicación sobre el ciclo de vida en proyectos de desarrollo de software.
- **7.5:** Se agrega la etapa de despliegue o implementación y sus prácticas asociadas.
- **8:** Se agrega la guía de Scrum para grandes proyectos
- Se agregaron gráficos en las distintas secciones del libro
- Se realizaron varias mejoras en las traducciones al español
- Se realizaron mejoras en las explicaciones globales del libro para hacerlo más entendible al lector

