



**DIGITAL TALENT**  
AGENCY

TEMA 2  
**MODELO AGILE**

**METODOLOGÍAS  
DE GESTIÓN DE PROYECTOS**

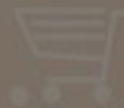


# ÍNDICE

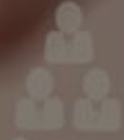
---

<b>TEMA 2   MODELO AGILE</b>	<b>3</b>
Introducción a Agile	<b>4</b>
El Manifiesto Agile	<b>5</b>
Principios del Manifiesto Agile	<b>5</b>
Metodología Kanban	<b>11</b>
Metodología Scrum	<b>15</b>

SECURITY



ONLINE SHOPPING



SOCIAL



GLOBAL

# TEMA 2 — MODELO AGILE



# INTRODUCCIÓN A AGILE

El concepto de Agile está cada vez más extendido, especialmente en el mundo del desarrollo de software, en el que se originó. Pero... ¿Qué es Agile? Se trata de una metodología de gestión de proyectos que adapta el trabajo en función de las necesidades finales del cliente y de las expectativas del usuario. Así, se consigue flexibilidad e inmediatez a la hora de adaptar el proyecto. Otra de las características de este método es que trata de transformar los entornos de trabajo para conseguir que los profesionales estén motivados y comprometidos a la hora de realizar sus labores.

La alta competitividad en el entorno de las Tecnologías de la Información ha provocado que los sistemas tengan que desarrollarse rápidamente, pero, a su vez, asegurando su calidad y la actualización constante de su código. Uno de los grandes errores que se cometen en este sentido es entregar proyectos superficiales o a medio terminar. Las metodologías ágiles nacen con el objetivo de evitar esto y asegurar la mejora continua de los servicios en los que se trabaja sin dejar de lado la rigurosidad ni los formalismos implícitos en las buenas metodologías de trabajo.

Aunque los frameworks más populares de Agile son Scrum, Kanban y XP, la verdadera clave de esta metodología es la transformación del comportamiento de los equipos, de su forma de pensar y trabajar. Por ello, la palabra Agile hace referencia a un cambio cultural profundo y no solo a la adopción de procesos y herramientas.

En relación con el capítulo anterior, Agile nació a causa de los problemas que empezaron a surgir con el Modelo en Cascada y, en definitiva, para solventarlos. Este modelo clásico no garantizaba la involucración del cliente a lo largo de todo el proceso de desarrollo, por lo que la comunicación era prácticamente nula. Los responsables del desarrollo tampoco participaban de forma activa en el proyecto y no había suficiente flexibilidad como para adaptarse a los potenciales cambios que pudieran surgir. El Modelo de Cascada exigía una gestión de documentación excesiva a lo largo del proceso, cosa que dilataba los tiempos de producción. Por último, esta falta de flexibilidad convertía a esta metodología en algo muy sensible a los errores.

En resumen, podemos decir que Agile permite mejorar la satisfacción del cliente, ahorrar costes en tareas tediosas y absurdas, hallar errores rápidamente, potenciar la eficiencia de los equipos y motivar a los profesionales implicados, cosa que se traduce en un producto de alta calidad.

# EL MANIFIESTO

El Manifiesto Agile fue acuñado el 17 de febrero de 2001 por 17 críticos de los modelos de desarrollo de software tradicionales basados en procesos, que ralentizaban el trabajo y se dilataban en el tiempo. Para encontrar una alternativa a estas metodologías formales, que dependían en gran medida de una planificación exhaustiva previa al desarrollo, idearon este manifiesto.

Los 17 autores crearon la web [agilemanifesto.org](http://agilemanifesto.org), en la que expusieron sus ideas. Es posible acceder a la **biografía de la mayoría de los fundadores** de la metodología de gestión de proyectos ágiles.

## PRINCIPIOS DEL MANIFIESTO AGILE

El Manifiesto Agile se basa en 4 grandes ideas, que se superponen a otras ya desfasadas:

- 1 - Individuos e interacciones** por encima de procesos y herramientas.
- 2 - Software funcional** por encima de documentación extensiva.
- 3 - Colaboración con el cliente** por encima de negociación contractual.
- 4 - Respuesta ante el cambio** por encima de seguir un plan.

# 1

## Individuo y sus interacciones por encima de procesos y herramientas

En muchas ocasiones no se tiene en cuenta que el motor del desarrollo de software son las personas, los profesionales. El Manifiesto Ágil pone al individuo por encima de los procesos y de las herramientas que utiliza puesto que, sin su trabajo, el proceso de desarrollo no sería posible.

Sin embargo, el individuo no es el único factor para tener en cuenta, ya que los equipos de trabajo son extremadamente valiosos. De poco sirve un profesional especializado si no coopera con el resto del equipo. El desarrollo de software, por ejemplo, requiere de una serie de profesionales que trabajen codo con codo y que se faciliten el trabajo entre sí. Las interacciones son otro elemento fundamental de esta metodología, ya que los equipos trabajan de forma conjunta y cotidiana, incluyendo a los responsables de negocio y a los desarrolladores. En este sentido, el Manifiesto Ágil trata de potenciar la comunicación cara a cara entre los profesionales, puesto que es lo más rápido y el mejor modo de que los trabajadores se entiendan.

El usuario final también forma parte, en cierto modo, del equipo de desarrollo, puesto que es el encargado de determinar si el trabajo es útil y si la experiencia es positiva. Por ello, su feedback es una de las bases del desarrollo.

Los entornos de trabajo condicionarán en gran medida el resultado final, por lo que deben adaptarse al equipo de profesionales y no al revés. Antaño, se construía el entorno de trabajo como primer paso y luego se forzaba a los desarrolladores a adaptarse a él. Las Metodologías Ágiles buscan satisfacer las necesidades de cada profesional permitiendo que el equipo configure el entorno de trabajo en función de sus características.

Si se tienen en cuenta todos estos factores, la elección de las herramientas a utilizar y los procesos pasan a un segundo plano, puesto que con un entorno de trabajo adaptado es altamente probable que se encuentren las herramientas indicadas con facilidad y que los protocolos nazcan con cohesión y aceptación general.

2

## Desarrollo de software funcional por encima de obtener documentación exhaustiva

Actualizar la documentación para poder trabajar en el desarrollo de sistemas es importante, pero teniendo en cuenta las presiones que existen en este sector es poco probable que los profesionales tengan tiempo para hacerlo.

La Metodología Ágil prioriza el desarrollo de software funcional libre de errores a la documentación continua, que puede provocar una pérdida de tiempo importante en el equipo. Básicamente, se trata de sacrificar el tiempo invertido en documentación innecesaria e implementarlo en el desarrollo como tal.

Lo más importante de este paso es no producir documentación superflua. Solo debería generarse aquella absolutamente primordial y que sea necesaria de forma inmediata para tomar decisiones vitales en el proceso de desarrollo. En definitiva, estos documentos deben ser escasos e ir al grano, ser breves, funcionales e inteligibles para todo el equipo. Lo más importante es, al fin y al cabo, desarrollar un software robusto y estable.





3

## Colaboración con el cliente por encima de la negociación de contratos

Para que un proyecto pueda alcanzar el éxito es importante que exista complicidad entre el cliente y el equipo de desarrollo. Como decíamos en el primer punto, el cliente debe sentir que forma parte del equipo, aunque no trabaje directamente sobre el propio desarrollo. Si esto se consigue, ambas partes comprenderán las necesidades de cada uno y podrán trabajar conjuntamente para optimizar el desarrollo.

Un contrato no establece la calidad o el valor del producto, sino que determina responsabilidades en el caso de disputas contractuales entre el cliente y el proveedor. Por ello, las Metodologías Ágiles impulsan las mejoras continuas con el objetivo de evitar que al final de todo el proceso el cliente quiera modificar algo y que eso quede fuera del presupuesto, cosa que inmovilizaría el proyecto por completo.



# 4

## Responder a los cambios por encima de seguir una planificación

Los proyectos deben adaptarse a los cambios que la propia organización sufre. Actualmente, las empresas mutan de forma constante, puesto que tratan de adaptarse al mercado. Por ello, es natural que el desarrollo de software sea un área dinámica y en evolución continua. Planificar el proyecto de desarrollo es importante, pero no hay que hacerlo de forma hermética y estricta, eliminando toda posibilidad de maniobra. La planificación debe ser abierta y flexible para que adaptar el proyecto a los cambios sea posible y sencillo.

El objetivo de esto es que, al terminar todo el proceso de desarrollo, el resultado siga siendo útil y no haya quedado obsoleto.



# 12 PRINCIPIOS AGILE

Además de estos cuatro valores, el Manifiesto Ágil contempla otros doce principios que definen las ventajas de las metodologías ágiles frente a las tradicionales:

1. Hacer entregas continuas de software para aportar valor al cliente.
2. Aceptar los cambios incluso al final del desarrollo para conseguir ventajas competitivas.
3. Entregar software funcional, entre cada dos semanas y dos meses, cuanto antes.
4. Los responsables del negocio y los desarrolladores deben trabajar juntos día a día.
5. Garantizar que el entorno de trabajo está adaptado a los desarrolladores para mantener motivado al equipo.
6. El diálogo cara a cara es fundamental para garantizar la comunicación en el equipo de desarrollo.
7. El software funcional marcará el progreso del proyecto.
8. El ritmo de trabajo debe ser constante y el desarrollo, sostenido.
9. La atención continua al diseño y a la calidad técnica mejoran la agilidad.
10. La simplicidad en los procesos de desarrollo es esencial.
11. Los mejores diseños y arquitecturas nacen de los equipos que se organizan y gestionan a sí mismos.
12. El equipo debe evaluar regularmente cómo ser más efectivo y, en función a esto, modifica su modo de proceder



# METODOLOGÍA KANBAN

## Introducción a Kanban

Kanban es un sistema basado en las Metodologías Ágiles. Este método se utiliza para asegurar la organización, productividad y eficiencia de los procesos, aunque también se puede aplicar a los departamentos en sí mismos. Se basa en el trabajo en equipo y en el flujo de tareas permanente. Como en Scrum, las tareas tienen una fecha límite, pero no es tan estricto, sino que ofrece cierta flexibilidad.

Kanban fue desarrollado por Taiichi Ohno, de Toyota, a principios de los años 40 para mejorar el sistema de producción Just in Time o JIT, que consiste en recibir cantidades justas y necesarias de suministros en la fábrica y justo en el momento en el que se van a utilizar para ahorrar costes en concepto de almacenamiento.

En sus orígenes, Kanban se utilizaba para controlar cómo avanzaba el trabajo en la línea de producción. Más tarde, se aplicó en el ámbito del desarrollo de software en Microsoft de la mano de David J. Anderson. Sin embargo, hoy en día se pueden aplicar los tableros Kanban para cualquier tipo de sector o ámbito, por lo que se trata de una metodología completamente democratizada.

Kanban basa su funcionamiento, como veremos, en tableros o pizarras en los que colocamos las tareas a realizar apuntadas en varios papeles. Esta metodología busca potenciar la visualización de tareas de forma rápida y sencilla para ahorrar tiempo en la planificación.

Veamos cómo desplegar una metodología Kanban:

# Organización Kanban

Los flujos de trabajo contemplados en los tableros Kanban deben contener tres elementos fundamentales para poder organizar el desarrollo de las tareas:

- **Pendientes.** Se trata de la lista de tareas que todavía no han sido empezadas.
- **En curso.** Tareas en las que se está trabajando en el momento. Las entregas que estén a punto de vencer deben marcarse como principales.
- **Realizadas.** Son las tareas ya finalizadas. Hay que mantenerlas en el tablero para tener un registro de lo que se ha hecho a lo largo de todo el proyecto.

Cabe destacar que las metodologías ágiles son flexibles y permiten modificar estos valores, por lo que podemos tener un tablero con estos tres valores y otros como “pruebas” o “ideas”, por ejemplo.





La metodología Kanban posee 4 principios básicos:

- **Visualización del flujo de trabajo y sus fases.** Kanban se basa en el desarrollo incremental, que divide el trabajo en varias secciones. Se utilizan técnicas visuales para comprobar en qué punto se encuentra cada tarea de forma rápida y sencilla. Cada una de las partes del flujo de trabajo se escribe en un post-it, por ejemplo, y se coloca en un lienzo o pizarra. En cada uno de estos post-it deberá incluirse la duración estimada de cada tarea y la descripción de lo que hay que hacer. Un ejemplo de herramienta que se utiliza en este sentido es Trello:

- **Limitar el trabajo en proceso.** En Kanban, el número máximo de tareas que pueden realizarse a la vez debe limitarse para garantizar el flujo de trabajo ágil. Por ello, en el propio tablero hay que establecer este número límite (ej. 3 tareas máximo). Además, en cada ciclo de desarrollo del proyecto deben establecerse también límites (ej. 3 tareas de desarrollo y 1 prueba). Por ello, detrás del nombre de cada tarea hay que escribir entre paréntesis este número máximo de tareas. La clave está en no abrir más tareas hasta que se hayan cerrado otras. Si hay un máximo de 3 tareas y se cierra 1 de 3, podremos abrir otra.

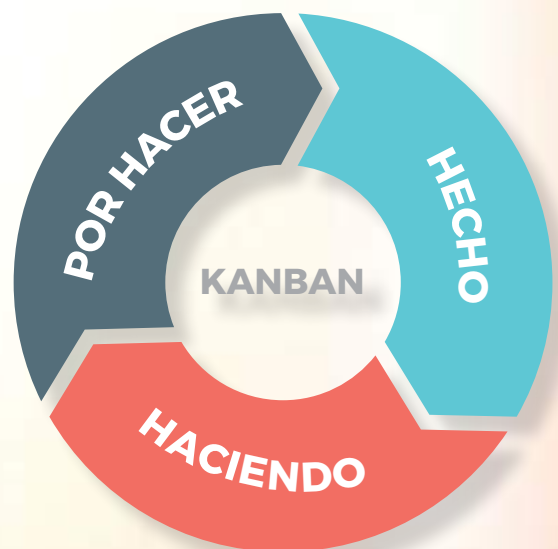
- **Priorización.** El bloque de tareas presentadas en el lienzo Kanban tiene que aportar transparencia y tiene que ser suficientemente visual como para comprobar rápidamente qué tareas necesitan priorizarse y cuáles están más avanzadas.

- **Medir el tiempo.** El Lead Time es el tiempo que pasa entre que se hace la petición de una tarea hasta que se cumple. Este concepto es fundamental en el ámbito de Kanban, pero también hay que tener presente el Cycle Time, que contempla desde el momento en el que se empieza una tarea hasta que se termina. El Lead Time está enfocado a clientes y el Cycle Time al propio equipo de desarrollo.

# Ventajas de utilizar Kanban

- Permite ver el trabajo en un solo vistazo.
- Se reduce o incluso se elimina por completo la ejecución de procesos innecesarios.
- Mejora el servicio que se le da al cliente gracias a las fechas pactadas en el tablero.
- Facilita mantener informado al equipo respecto al proceso que están desarrollando.
- Alinea al equipo con un objetivo muy concreto y potencia su motivación.
- Acepta todos los cambios necesarios, aunque se reciban en mitad del desarrollo de una tarea.
- Optimiza el tiempo y mejora la eficiencia de los profesionales.

En conclusión, lo que Kanban trata de lograr es garantizar el flujo de trabajo para que nunca se detenga. Cuando se termina una tarea debe empezarse otra, pero jamás debe rebasarse el límite de tareas establecido por el equipo. De este modo, trabajando tarea a tarea, se evitarán errores y se centrará el trabajo en tareas concretas. El flujo debe seguir la lógica:



# Metodología Scrum

## Qué es Scrum y por qué deberíamos utilizarlo

Un proyecto Scrum se basa en el trabajo colaborativo para crear un nuevo producto o servicio. Se trata de un framework adaptable, iterativo, rápido, flexible y eficaz que ofrece una mayor velocidad comparado con otras metodologías. Además, Scrum también garantiza una mayor transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo.

Scrum puede utilizarse en el desarrollo de productos y soluciones para cualquier tipo de industria y en cualquier tipo de proyecto, sea cual sea su complejidad. La clave es, precisamente, que los equipos Scrum son interfuncionales, autoorganizados y empoderados, tal y como veremos en los siguientes apartados. Estos equipos trabajan de forma colaborativa dividiendo las partes del proyecto en ciclos de trabajo cortos y concentrados llamados sprints.

La metodología Scrum siempre trabaja con un mismo ciclo o flujo de trabajo, que detallaremos más adelante. Este ciclo se ha ido perfeccionando con los años, desde que se presentó en 1995 de la mano de Ken Schwaber y Jeff Sutherland, que se basaron a su vez en las estrategias que definieron Hirotaka Takeguchi e Ikujiro Nonaka en la década de 1980. En un primer momento, se describió el método Scrum como el desarrollo flexible de un producto donde el equipo trabaja en unidad para alcanzar un objetivo común. Para ilustrarlo, podemos pensar en que el desarrollo de productos no debe ser como una carrera de relevos, sino algo parecido a un juego de rugby, donde los jugadores se van pasando el balón hacia atrás y hacia adelante a medida que el equipo en conjunto se desplaza.

¿Cuáles son las **ventajas principales del Scrum**? Existen muchos motivos para utilizar la metodología Scrum, pero las claves principales podrían ser resumidas en:

- Adaptabilidad
- Transparencia
- Mejora y retroalimentaciones continuas
- Ritmo sostenible
- Proceso de desarrollo eficiente
- Motivación
- Resolución de problemas de forma rápida
- Centrado en el cliente
- Ambiente de alta confianza
- Responsabilidad colectiva
- Ambiente innovador

# Principios de Scrum

El framework Scrum cuenta con 6 principios o pautas básicas que deben aplicarse obligatoriamente y pueden modificarse para cumplir con los requisitos del proyecto en cuestión:

**1. Control del proceso empírico** (Empirical Process Control). Filosofía central de Scrum, basada en las tres claves principales de la metodología: transparencia, inspección y adaptación.

**2. Autoorganización** (Self-organization). Equipos autoorganizados, que cuentan con un gran sentido del compromiso y la responsabilidad que ayuda a potenciar la innovación.

**3. Colaboración** (Collaboration). Se fomenta la gestión de proyectos que tengan valor compartido y equipos que interactúen continuamente.

**4. Priorización basada en valor** (Value-Based Prioritization). Ofrecer el máximo valor de negocio, de principio a fin del proyecto.

**5. Time-Boxing.** Los principales elementos que afectan a la gestión del tiempo en Scrum son: sprints, Dalily Standups, reuniones de planificación del sprint y reuniones de revisión del sprint.

**6. Desarrollo iterativo** (Iterative Development). Se describen las responsabilidades del Product Owner y las de la organización en relación con el desarrollo iterativo del proyecto.







## Organización de equipos Scrum

- **Product Owner.** Es el responsable de lograr el máximo valor empresarial y de articular los requisitos del cliente.
- **Scrum Master.** Es un facilitador que se asegura de que exista un ambiente propicio para que el equipo complete el proyecto con éxito. Es un guía que enseña las prácticas Scrum a todas las personas involucradas en el proyecto para así eliminar los impedimentos que puedan ir surgiendo.
- **Equipo Scrum.** Es el grupo de personas responsables de entender los requisitos que ha especificado el Product Owner y de crear los entregables del proyecto.
- **Stakeholders.** Incluye a clientes, usuarios y patrocinadores, que suelen interactuar con el equipo principal de Scrum e influyen en el proyecto.
- **Scrum Guidance Body.** Se trata de un rol opcional que engloba a un grupo de expertos que normalmente están involucrados en la definición de objetivos relacionados con la calidad, las regulaciones gubernamentales, la seguridad y otros parámetros necesarios para el proyecto.
- **Vendedores.** Incluye individuos u organizaciones externas que ofrecen servicios que no están dentro de las competencias centrales del equipo que se ocupa del proyecto.



# Flujo de Trabajo de un proyecto Scrum

## 1. INICIO

**1.1 Creación de la visión del proyecto.** En esta primera fase se revisa el caso de negocio del proyecto (o Project Business Case) con el objetivo de crear una Declaración de la visión del proyecto, que servirá como punto de partida y proporcionará un enfoque para el conjunto del proyecto. Durante este proceso se identifica el Product Owner.

**1.2 Identificación del Scrum Master y los Stakeholders.** Se utilizan criterios de selección específicos para identificar al Scrum Master y a los stakeholders.

**1.3 Formación de equipos.** Normalmente, el Product Owner es la persona responsable de identificar a los miembros del equipo.

**1.4 Desarrollo de épica(s).** La Declaración de visión del proyecto se utiliza como base para el desarrollo de épicas o conjuntos de historias de usuario. Las historias de usuario son las funcionalidades que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente.

**1.5 Creación del Backlog Priorizado del Producto.** Se crean las épicas y se priorizan para crear un Backlog Priorizado del Producto para el proyecto.

**1.6 Realización de la planificación del lanzamiento.** El equipo Scrum revisa la historia de usuario en el Backlog Priorizado del Producto para desarrollar un cronograma del lanzamiento, que es un programa de implementación por fases. Se determina también la duración del sprint.

## 2. PLANIFICACIÓN Y ESTIMACIÓN

**2.1 Creación de historias de usuario.** Se crean las historias de usuario y los criterios de aceptación de las historias de usuario. Normalmente, las historias de usuario son escritas por el Product Owner y se diseñan para asegurar que los requisitos del cliente estén claramente representados y sean comprensibles para todos los stakeholders. Se pueden crear las historias de usuario y se incorporan al Backlog Priorizado del Producto.

**2.2 Estimación de historias de usuario.** Se estima el esfuerzo y tiempo necesario para desarrollar la funcionalidad de cada historia de usuario.

**2.3 Compromiso de historias de usuario.** El equipo Scrum se compromete a entregar al Product Owner las historias de usuario aprobadas para un sprint.

**2.4 Identificación de tareas.** Las historias de usuario se dividen en tareas específicas.

**2.5 Estimación de tareas.** Se estima el esfuerzo y tiempo necesario para cumplir cada una de las tareas, también llamado Effort Estimated Task List.

**2.6 Creación del Sprint Backlog.** El equipo Scrum elabora un Sprint Backlog que contiene todas las tareas que se deben llevar a cabo en un sprint como parte de la Reunión de Planificación del Sprint.

## 3. IMPLEMENTACIÓN

**3.1 Creación de entregables.** El Equipo Scrum trabaja en el Sprint Backlog para crear los entregables del sprint. Los problemas o retos que puede ir encontrando el equipo Scrum pueden actualizarse en un registro de impedimentos o Impediment Log.

**3.2 Realización del Daily Standup.** Se lleva a cabo una reunión diaria, también llamada Daily Standup, donde los miembros del equipo Scrum informan al resto sobre sus progresos y los problemas que le han surgido.

**3.3 Actualización del Backlog Priorizado del Producto.** Se analiza cualquier cambio o actualización al backlog.

## 4. REVISIÓN Y RETROSPECTIVA

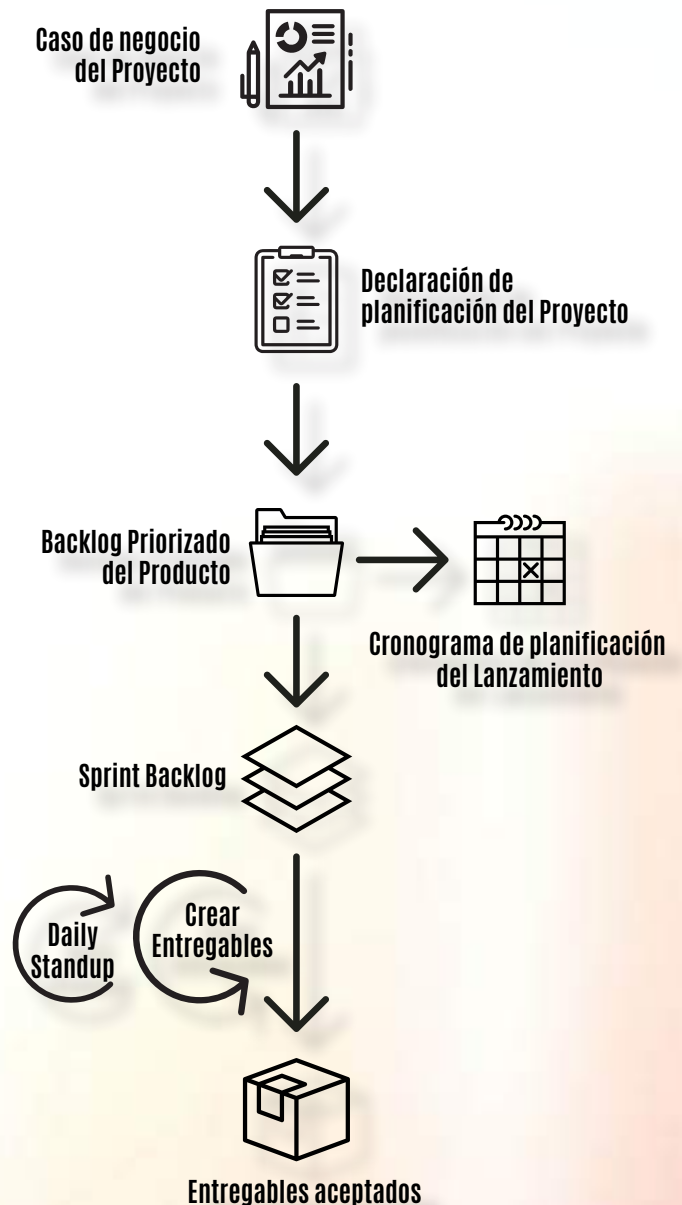
**4.1. Validación del sprint.** El equipo Scrum muestra los entregables del sprint al Product Owner y a los stakeholders relevantes en una Reunión de Revisión del Sprint para que sean aprobados.

**4.2 Reunión retrospectiva del sprint.** El Scrum Master y el equipo Scrum se reúnen para hacer una retrospectiva de todo el sprint. En este punto, es importante documentar todo lo aprendido para aplicarlo a futuros sprints y se pueden aprobar algunas mejoras o recomendaciones (Agreed Actionable Improvements).

## 5. LANZAMIENTO

**5.1 Envío de entregables.** Los entregables aceptados se envían a los stakeholders y se documenta la conclusión del sprint a través del acuerdo de entregables funcionales (Working Deliverables Agreement).

**5.2 Realización de reunión retrospectiva.** En este punto, en el cual el proyecto se da por finalizado, los stakeholders y los miembros del equipo Scrum se reúnen para hacer una retrospectiva del proyecto y documentar lo aprendido.





**zemsania**  
Global Group<sup>®</sup>

EXPERTOS EN TALENTO TIC & DIGITAL



**DIGITAL TALENT**  
**AGENCY**

*El punto de encuentro digital  
entre talento y empresas*

[www.dtagency.tech](http://www.dtagency.tech)