# Metodologías

Las metodologías de desarrollo de software son un conjunto de técnicas y métodos que se emplean para estructurar, planificar y controlar el proceso de desarrollo de software. En segundo lugar, filosofías o políticas que pueden analizar el modo en el que pretendemos adaptar tecnología con la definición de soluciones y que pueden dar lugar al desarrollo de las herramientas y técnicas que pueden elaborar una solución.

## Metodologías Agiles

Las metodologías ágiles se fundamentan en el manifiesto ágil, creado en 2001 por un grupo de desarrolladores de software que buscaban una forma más efectiva de desarrollar el software, el cual tiene como principios:

1. Individuos e interacciones sobre procesos y herramientas: Valorar más a los individuos y sus interacciones que a los procesos y las herramientas.
2. Software funcionando sobre documentación extensiva: el objetivo principal es proporcionar software funcional.
3. Colaboración con el cliente sobre negociación contractual: la comunicación constante con el cliente es esencial para asegurar el producto final.
4. Respuesta ante el cambio sobre seguir un plan: la capacidad de adaptarse y responder a los cambios es mejor que seguir un plan fijo.

Normalmente las metodologías agiles se basan en un desarrollo incremental, lo cual se traduce en un software entregable que puede ser explotado. Este tipo de desarrollos nos permiten entregar pocas funcionalidades, pero evidentes y fácilmente adaptables.

Las metodologías agiles permiten a los equipos de desarrollo adaptarse rápidamente a los cambios en los requisitos y prioridades del cliente, mejorar la colaboración y comunicación y entregar productos de calidad de manera continua.

Se explicará en detalle las metodologías más importantes, Scrum y Kanban.

## Scrum

Framework de desarrollo ágil con el fin de administrar proyectos complejos. Fomenta la colaboración, flexibilidad y la entrega incremental de productos y centrado en la mejora continua.

Scrum está concebido con el propósito de asistir a los equipos a adaptarse de manera natural a las circunstancias y exigencias de los usuarios, mediante el cambio de prioridades integrado en el proceso y los ciclos cortos para que el equipo pueda adquirir y mejorar constantemente.

Detallaremos los roles de un equipo de Scrum y sus responsabilidades, los artefactos que definen el producto y el trabajo que hay que hacer para crear el producto.

Roles:

Product Owner: es el representante del cliente, y se centra en entender los requisitos empresariales de los clientes y mercado.

Scrum Master: facilita el proceso de Scrum, ayuda a eliminar obstáculos y asegura que el equipo siga las prácticas de Scrum.

Equipo de desarrollo: grupo de desarrolladores que trabajan en los incrementos del producto.

Eventos:

Sprint: un ciclo de trabajo corto y repetitivo, típicamente de 2 a 4 semanas en el que se crea un incremento del producto utilizable.

Planificación de Sprints: reunión dirigida por el Scrum Master donde el equipo decide el objetivo y alcance del Sprint actual. Además, se añaden al Sprint historias de usuario especificas a partir del backlog del producto.

Scrum diario: reunión diaria, en la que el equipo lleva a cabo una planificación temporal de las actividades correspondientes para las siguientes 24 horas.

Revisión del Sprint: reunión final del Sprint en el que el equipo presenta el resultado al Product Owner.

Artefactos:

Product backlog: Se trata de la lista de tareas principales que se deben llevar a cabo. debe mantener el Product Owner a medida que crece y evoluciona durante el desarrollo del producto. A estos trabajos se les denomina historias de usuario.

Product backlog: Se trata de la lista de tareas principales que se deben llevar a cabo. Esta lista de tareas deberá ser mantenida por el Product Owner a medida que evoluciona el proyecto.

Sprint Backlog: se trata de una lista de elementos e historias de usuario seleccionadas por el equipo de desarrollo con el fin de implementarlos en el Sprint actual.

Incremento: la suma de todos los elementos del producto backlog completados durante el sprint actual.

Épicas: se trata de cantidades de trabajo que se pueden desglosar en un número de tareas más pequeñas.

## Kanban

Esta metodología ágil es utilizada para gestionar y mejorar el trabajo. Requiere una comunicación en tiempo real sobre la capacidad y una total transparencia del trabajo.

Es una forma de visualizar el trabajo y limitar la cantidad de trabajo en curso para que el equipo pueda lograr un flujo de trabajo eficiente.

En esta metodología se hace uso de un tablero de Kanban, que se utiliza para visualizar el flujo de trabajo entre los equipos y garantiza que se visualice el trabajo realizado, estandarice su flujo de trabajo, y se identifiquen y resuelvan todos los impedimentos. Estos tableros tienen un flujo básico de tres pasos:

Por hacer (To do): tareas pendientes.

En progreso (In progress): tareas que se están realizando en un momento determinado.

Hecho (Done): tareas finalizadas.

## Scrumban

La metodología Scrumban combina las mejores características de Scrum y Kanban en un framework de gestión de proyectos hibrido. Utiliza el flujo de trabajo visual del Kanban y la estructura estable de Scrum.

Scrum contribuye a Scrumban en los elementos principales de Sprints y reuniones rápidas diarias.

Kanban contribuye a Scrumban con el tablero de Kanban y las tarjetas representantes de las tareas del proyecto y su estado.

## Metodología escogida (Scrumban) SIN CHECKAR EL 4 APARTADO

En el proyecto se utiliza la metodología ágil Scrumban mediante el uso de las herramientas que nos proporcionan las metodologías anteriores, Kanban y Scrum.

Utilizaremos una herramienta del tablero de Kanban digital, para la cual hay una gran variedad de herramientas disponibles. Se optó por Taiga ya que es una herramienta Open-Source enfocada en el uso de elementos Kanban y Scrum.

Además del tablero Kanban digital, también utilizo uso de las historias de usuario derivadas de las funcionalidades a implementar especificadas durante el análisis, así como las épicas.

El proyecto se estructura en distintos Sprints, en los que se realizarán entre 1 o 2 épicas en función de la cantidad de historias de usuario. Se realizará una evaluación del diseño al comienzo de cada Sprint, aunque el primer Sprint consistirá en el diseño de una primera versión de la capa modelo y servicios, y también se elaborarán prototipos de las pantallas del aplicativo móvil.

Dado que se trata de un proyecto llevado a cabo por una persona no se podrán aplicar ciertas prácticas de colaboración entre miembros de un equipo de desarrollo. Sin embargo, se empleará un sistema de puntuación que proporciona la herramienta para asignar “puntos historia” a cada una de las historias de usuario, lo que nos permitirá visualizar el nivel de esfuerzo requerido para la ejecución de las tareas.