**Modelo Scrum**

Este modelo se base en aspectos como la flexibilidad en adopción a cambios y nuevos requisitos durante un proyecto complejo, el factor humano, la colaboración e interacción con el cliente y el desarrollo iterativo como formas de asegurar buenos resultados.

Los pilares más importantes de la metodología Scrum son:

* **Transparencia:** todos los implicados tienen conocimiento de que ocurre en el proyecto.
* **Inspección:** los miembros del equipo Scrum frecuentemente inspeccionan el progreso para detectar posibles problemas.
* **Adaptación:** cuando hay algo que cambie el equipo se ajusta para conseguir el objetivo del sprint.

**Roles de Scrum**

* **Product owner:** es el responsable de maximizar el valor del trabajo del equipo de desarrollo, es la única persona que habla constantemente con el cliente, la persona puede ser parte del equipo de desarrollo.
* **Scrum master:** es la persona responsable de que Scrum sea comprendido y aplicado en la organización, se encarga de eliminar impedimentos o inconvenientes que tenga el equipo dentro de un sprint, aplicando las mejores técnicas para fortalecer el equipo de marketing digital.
* **Equipo de desarrollo:** son las personas encargadas de realizar las tareas priorizadas por el product owner, es un equipo multifuncional y auto-organizad, son los únicos que estiman las tareas del product backlog.

**Eventos del scrum**

1. **Sprint:** es cada uno de los ciclos o iteraciones que vamos a tener dentro de un proyecto, la duración máxima del esprint es de 1 mes, el tiempo se determina en base al nivel de comunicación que el cliente quiere tener con nosotros. Cada sprint es un entregable o incremento del producto.
2. **Sprint plannig:** es la primera reunión que se hace en el sprint, en esta reunión todo el equipo Scrum define que tareas se van a abordar y cuál será el objetivo del sprint. Se realizan dos preguntas **que se va a hacer y cómo ser hará.** A los entregables se les conoce como **Sprint goal**
3. **Daily meeting:** es una reunión diaria dentro del Sprint que tiene máximo como 15 minutos de duración, no es necesario que el Scrum master o product owner este en esa reunión
   1. **Que hice ayer**
   2. **Que voy hacer hoy**
   3. **Tengo algún impedimento que necesito que me solucionen**
4. **Sprint review:** su duración es de 4 horas para Sprint de un mes, es la reunión donde asiste el cliente, donde el producto owner presenta lo desarrollado al cliente y el equipo de desarrollo muestra su funcionamiento, donde en esta reunión se validan los cambios realizados y se generan feedback sobre nuevas tareas que el producto owner tendrá que agregar al product backlog.
5. **Sprint retrospective:** La retrospectiva es el último evento de Scrum, tiene una duración de 3 horas para Sprints de 1 mes, y es la reunión del equipo en la que se hace una evaluación de cómo se ha implementado Scrum en el finalizado Sprint.

**El resultado**: una lista de mejoras que debe aplicar el siguiente día, ya que al finalizar la retrospectiva, inmediatamente comienza un nuevo Sprint, que incluye el Sprint Planning, Daily Meeting, Sprint review y la ya mencionada Sprint Retrospective.

**Artefactos del Scrum**

Los artefactos de Scrum están definidos para maximizar la transparencia dentro del equipo, es decir, que todos tengan una misma visión de lo que hay en el proyecto.

* **Product Backlog**

Básicamente, el Product Backlog es el **listado tareas que engloba todo un proyecto**. Cualquier cosa que debamos hacer debe estar en el Product Backlog, estimado por el equipo de desarrollo.

La responsabilidad exclusiva de ordenar el Product Backlog es del Product Owner, que se encuentra en constante comunicación con el cliente para asegurarse de que las prioridades están bien establecidas.

La ordenación también es 100% responsabilidad del Product Owner, **donde las tareas que están más arriba son las de mayor prioridad.**

* **Sprint Backlog**

Es el grupo de tareas del Product Backlog que el equipo de desarrollo elige en el Sprint Planning junto con el plan para poder desarrollarlas. Debe ser conocido por todo el equipo, para asegurarse que el foco debe estar en este grupo de tareas.

El Sprint Planning **no cambia durante el sprint**, solo se permite cambiar el plan para poder desarrollarlas.

**Que engloba un srpint**

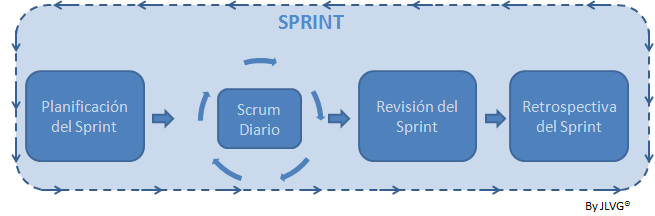
Cuando hablamos de Sprint en Scrum técnico, engloba todo el proceso, es decir, **desde que decidimos qué vamos a hacer para ese Sprint, hasta que estudiamos cómo hemos trabajando en ese Sprint**.

Cuando estamos en un proyecto Scrum y comenzamos un Sprint, el mismo siempre **comienza con la reunión de planificación del Sprint**. En esa reunión vamos a decidir lo que vamos a hacer y cómo lo vamos a hacer, el número de tareas o de historias de usuario que vamos a realizar en el Sprint.

Después vamos a tener otra serie de reuniones:

* **Reuniones de Scrum diario**, que van a ser pequeñas reuniones con los miembros del equipo.
* **Revisión del Sprint**, en la que vamos a aceptar o denegar el Sprint.
* **Reunión de retrospectiva**, dónde vamos a ver cómo ha trabajado el equipo y qué problemas ha tenido durante el desarrollo y cómo lo podemos corregir.

**El Spring engloba todo lo anterior**, desde que comienza el mismo hasta que es aceptado o denegado, y el equipo se pregunta cómo ha trabajado.



**Git-Flow**

Es una forma ordenada de manejar las ramas y repositorios para poder gestionar de mejor forma un proyecto. La clave de es esto es generar una rama y que esta rama no afecte a los demás.

* Creamos nuestro repositorio.
* Descargamos el repositorio (clonamos) desde la línea de comandos.
* Creamos la rama develop (esto es para que no se trabaje en la rama master)
* De la rama develop creamos las ramas para los integrantes de cada grupo de desarrollo (por cada rama que se haga realizar un push)
* Cada desarrollador tiene la posibilidad de crear su propia rama.
* Hacer el commit por cada cambio que se realice en su respectiva rama
* Hacer el merge branch a la rama develop (esto lo hace cada desarrollador desde su rama)
  + Cuando el siguiente desarrollador hace el Marge debe de tener actualizada la rama
  + Solucionar los conflictos indica que hay que hacer otro commit

Base de Git-flow

* La rama principal (master) solo tiene código de producción.
* En la rama de desarrollo (develop) solo se guarda código que ya está probado y es funcional.
* El trabajo se organiza en funcionalidades (features) completas. Por ejemplo: «AMI de nagios», «login de usuarios», «etc.
* Junto con las funcionalidades, podemos tener cambios urgentes (hotfix), que solucionen problemas urgentes.
* Por último, tendremos versiones (releases), que son un grupo de características funcionales (por ejemplo, versión 1.1.4 o 2.0)