**CAPITULO 4- METODOLOGIA DE DESARROLLO.**

SCRUM: es un marco de trabajo para la gestión y [desarrollo de software](http://es.wikipedia.org/wiki/Desarrollo_de_software) basada en un proceso [iterativo e incremental](http://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente) utilizado comúnmente en entornos basados en el [desarrollo ágil de software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software).

En SRUM hay dos aspectos fundamentales a diferenciar, los actores y las acciones. Los actores son los que ejecutarán las acciones y está formado por las siguientes personas:

**Product Owner:** conoce y marca las prioridades del proyecto o producto.

**Scrum Master:** es la persona que asegura el seguimiento de la metodología guiando las reuniones y ayudando al equipo ante cualquier problema que pueda aparecer. Su responsabilidad es entre otras, la de hacer de paraguas ante las presiones externas.

**Scrum Team:** son las personas responsables de implementar la funcionalidad o funcionalidades elegidas por el Product Owner.

**Usuarios o Cliente:** son los beneficiarios finales del producto, y son quienes viendo los progresos, pueden aportar ideas, sugerencias o necesidades.

Para el proyecto de tesis, nuestro actores son:

|  |  |
| --- | --- |
| SCRUM | |
| **Product Owner** | Dueño de restauran |
| **Scrum Master** | Profesora |
| **Scrum Team** | Martin y Walter |
| **Usuarios o Cliente** | Amigos de la profe. |

**Las accione:**

**Product Backlog:** Corresponde con todas las tareas, funcionalidades y requerimientos a realizar. El **Product Owner** es la persona que se encarga de marcar las prioridades y de mantener y actualizar la lista de tareas.

**Sprint Backlog**: Corresponde con una o más tareas que provienen del Product Backlog. El Sprint Backlog tiene una duración de 2 a 4 semanas. Mientras en Spring Backlog se inicia, este no puede ser alterado o modificado. Se tiene que esperar a que concluya el Spring Backlog para realizar la correspondiente modificación o alteración cuya tareas, formaría para de otro Spring Backlog. Cuando se ha finalizado un Sprint Backlog, se debe tener un entregable o algo que se pueda mostrar y que enseñe los avances acometidos en el Sprint.

**Daily Scrum Meeting:** Es una tarea iterativa que se realiza todos los días que dure el Sprint Backlog con el equipo de desarrollo o de trabajo. Es una reunión operativa, informal y ágil, de un máximo de 30 minutos, en la que se hace 3 preguntas a cada integrante del equipo:

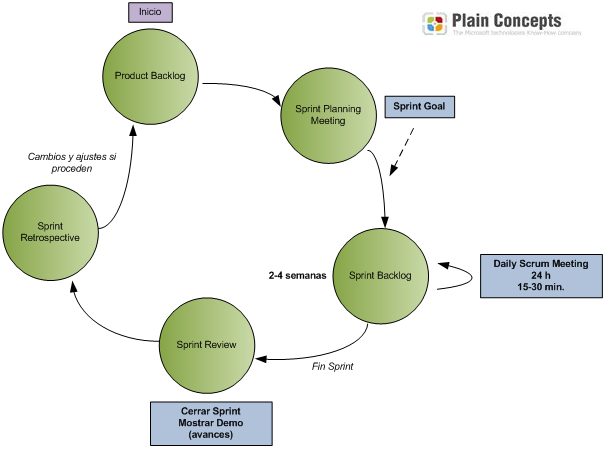
* Que tareas ha realizado desde la última reunión (que he hecho).
* Sobre qué va a trabajar en el día actual (que voy a hacer hoy).
* Identificación de obstáculos o riesgos que impiden o pueden impedir el normal avance (que ayuda necesito). El Scrum Master debe eliminar cualquier obstáculo que encuentre.

**Sprint Planning Meeting**: es una reunión que tiene por objetivo, planificar el Sprint a partir del **Product Backlog.** El objetivo de esta reunión es la de mover las tareas del **Product Backlog** al **Sprint Backlog**. En esta reunión, suelen participar el **Product** **Owner** que es quien prioriza las tareas.

**Sprint Goal**: es un pequeño documento o una breve descripción que indica lo que el **Sprint** **Backlog** intentará alcanzar.

**Sprint Review:** se revisa en unas 2 horas como máximo el Sprint finalizado. Al llegar a este punto, debemos tener "algo" que el Cliente o el Usuario pueda ver y tocar. En esta reunión, suelen asistir el **Product Owner**, el **Scrum Master**, el **Scrum Team** y personas que podrían estar involucradas en el proyecto. El **Scrum Team** es quién muestra los avances realizados en el Sprint.

**Sprint Retrospective**: Al finalizar un Sprint Backlog y el Sprint Review, se inicia el **Sprint Retrospective.** El Product Owner revisará con el equipo los objetivos marcados inicialmente en el Sprint Backlog concluido, se aplicarán los cambios y ajustes si son necesarios, y se marcarán los aspectos positivos (para repetirlos) y los aspectos negativos (para evitar que se repitan) del Sprint.

****

**Planning poker** : es una técnica para calcular una estimación basada en el concenso, en su mayoría utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de [desarrollo de software](http://es.wikipedia.org/wiki/Desarrollo_de_software).

El póker de planeamiento está basado en una lista de tareas para ser entregados y una baraja de cartas. Las cartas en el mazo están numeradas. Un mazo típico contiene tarjetas mostrando la secuencia de Fibonacci incluyendo un cero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Otros mazos utilizan progresiones similares. La razón de utilizar la secuencia de Fibonacci es reflejar la incertidumbre inherente en la estimación. Un mazo que se encuentra en el mercado utiliza la siguiente secuencia: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, y, opcionalmente, una tarjeta

**Procedimiento**

En la reunión de la estimación a cada estimador se le da un conjunto completo de tarjetas.

La reunión prosigue de la siguiente manera:

* Un moderador, que no jugará, preside la reunión, apoyado y asesorado por el Gestor del Proyecto.
* El desarrollador con más conocimiento de una determinada característica proporciona una breve introducción sobre la misma. El equipo tiene la oportunidad de hacer preguntas y discutir para aclarar los supuestos y riesgos. Un resumen de la discusión es registrado por el Gestor del Proyecto.
* Cada persona coloca una tarjeta boca abajo que representa su estimación. Las unidades utilizadas pueden ser variadas y definidas previamente. Pueden ser días de duración, días ideales o puntos de la historia. Durante el debate, los números no debe ser mencionados en absoluto.
* Todo el mundo muestra sus tarjetas de forma simultánea.
* A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa.
* Se repita el proceso de cálculo hasta que se alcance un concenso. El programador que probablemente tenga el entregable tiene una gran parte del voto de concenso, aunque el moderador puede negociar el concenso.
* Se puede utilizar un reloj de arena para asegurar que el debate sea estructurado, el moderador o el Gestor del Proyecto podrá en cualquier punto terminar el reloj y cuando se acaba toda discusión debe cesar y otra ronda de póker se juega.

Las cartas están numeradas de esta forma para explicar el hecho de que, cuanto una estimación es mayor, existe mayor incertidumbre. Así, si un desarrollador quiere jugar un 6 se ve obligado a reconsiderar y aceptar que parte de la incertidumbre percibida no existe y jugar un 5, o aceptar una estimación más conservadora de la incertidumbre y jugar un 8.