

Actividad Scrum

1. Un cliente quiere desarrollar una nueva red social de juegos de mesa y para ello se ha puesto en contacto con nosotros.



¿Qué va a querer el cliente?

El cliente quiere una red social multiplataforma para móvil, centrada en juegos de mesa. Sus funcionalidades clave se dividen en prioridades:

Alta: Red social (Muro, perfiles, chat, clanes), geolocalización para quedadas y sistema de ELO/Ranking.

Media: Gestión de partidas físicas, torneos y ligas (incluyendo pagos y moderación).

Baja: Un simulador de mesa virtual.

2. El cliente se va a reunir con un miembro del equipo.

La tarea de esta persona consistirá en tomar nota de lo que el cliente necesita para ajustar el producto final lo más posible a la idea del cliente.

¿Con qué miembro del equipo se reunirá?

El cliente se reúne con el Product Owner. En Scrum, el Product Owner es la "voz del cliente" dentro del equipo y el encargado de gestionar el producto.

¿Cuál será el resultado de esta reunión?

El resultado es la creación de la Visión del Proyecto y el borrador inicial del Product Backlog (la lista de requisitos). El Product Backlog es el núcleo del trabajo.

3. Lo siguiente será realizar una división del producto, con el objetivo de desarrollar el Product Backlog. Un ejemplo podría ser este:

PRODUCT BACKLOG EXAMPLE						
ID	As a...	I want to be able to...	So that...	Priority	Sprint	Status
1	Administrator	see a list of all members and visitors	I can monitor site visits	Must	1	Done
2	Administrator	add new categories	I can allow members to create engaging content	Must	1	Done
3	Administrator	add new security groups	security levels are appropriate	Must	1	Done
4	Administrator	add new keywords	content is easy to group and search for	Must	1	Done
5	Administrator	delete comments	offensive content is removed	Must	1	Done
6	Administrator	block entries	competitors and offenders cannot submit content	Must	1	Done
7	Administrator	change site branding	the site is future-proofed in case brand changes	Could	1	Done
8	Member	change my password	I can keep secure	Must	1	Done
9	Member	update my contact details	I can be contacted by Administrators	Must	2	Work in Progress
10	Member	update my email preferences	I'm not bombarded with junk email	Should	2	Work in Progress
11	Member	share content to social networks	I can promote what I find interesting	Could	2	Work in Progress
12	Visitor	create an account	I can benefit from member discounts	Must		To be started
13	Visitor	login	I can post new entries	Must		To be started
14	Visitor	add comments	I can have a say	Must		To be started
15	Visitor	suggest improvements	I can contribute to the site usability	Should		To be started
16	Visitor	contact the Administrators	I can directly submit a query	Could		To be started
17	Visitor	follow a member's updates	I'm informed of updates from members I find interesting	Should		To be started
18	Visitor	view a member's profile	I can know more about a member	Must		To be started
19	Administrator	generate incoming traffic report	I can understand where traffic is coming from	Must		To be started

Una posible tabla que te presento yo es esta, la cual está basadas en **historias de usuario**. ¿Qué es eso?

La historia de usuario es básicamente una explicación breve y sencilla de la funcionalidad del software, pero desde la perspectiva del usuario y no desde la perspectiva técnica del programador.

En lugar de escribir requisitos técnicos, en Scrum explicamos cuál es la necesidad que tiene el usuario para que haga falta tal requisito. Lo más importante es saber qué problemas se resuelven con este requisito.

	As a...	I want to be able to ...	So that ...	Priority	Time	Sprint	Status

Vemos que he añadido una columna adicional llamada “Time”, esta columna mostrará una estimación de tiempo. ¿Qué técnicas podemos utilizar para establecer esta estimación de tiempo?

Podemos usar la técnica de planning poker que consiste en que el equipo use una baraja de cartas con la secuencia de Fibonacci para votar el esfuerzo de cada tarea.



4. En nuestro proceso de planning póker (<https://planningpokeronline.com/>) vamos a utilizar una baraja que tiene las siguientes cartas 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100 y además, dos tarjetas: una con signo de interrogación (?) y otra con signo de café (∞).

El procedimiento será el siguiente:

1. Un moderador, que no jugará, preside la reunión apoyado y asesorado por el Gestor del Proyecto.
2. 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.
3. 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 deben ser mencionados en absoluto.
4. Todo el mundo muestra sus tarjetas de forma simultánea.
5. A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa.
6. Se vuelve a realizar el mismo proceso.
7. Se suma la puntuación nueva y se añade como estimación de tiempo.

5. También deberemos definir las prioridades de cada tarea y el sprint al que pertenecerá. ¿De cuánto tiempo será como mínimo y como máximo un Sprint? Aquí hay una serie de técnicas para priorizar tareas:

<https://www.atsistemas.com/blog/tcnicas-de-priorizacin-el-desafo-de-conseguir-un-orden-para-las-funcionalidades>

6. ¡Por fin comenzamos el desarrollo! Comenzaremos con una reunión de planificación del sprint ¿cómo se llama? Una de las actividades que debemos hacer es dividir el Sprint en actividades más pequeñas (divide y vencerás)

La reunión de planificación se llama Sprint Planning Meeting, donde el equipo coge las historias de usuario que hicimos en el product backlog y las divide en tareas mucho más pequeñas que se llaman “tasks”.

Por ejemplo:



Historias de usuario	Tasks
Como usuario, poder iniciar sesión para guardar mi proceso	1. Diseñar el esquema de la base de datos de usuarios
	2. Función de hash seguro para contraseñas
	3. Pantalla de Login/Registro
	4. Validación de credenciales

7. El resultado del sprint planning va a ser el Sprint Backlog. Para su gestión podemos utilizar la técnica de Kanban. ¿Cómo quedaría un posible tablero?

Sería más o menos así:

Por Hacer (To Do): Son las tareas que hemos sacado del *Sprint Planning* pero que nadie ha empezado todavía.

En Proceso (In Progress): El desarrollador coge una tarea (por ejemplo, "Maquetar Login") y la mueve aquí para indicar que está trabajando en ella.

Importante: Se suele limitar la cantidad de tareas aquí (WIP - Work In Progress) para no saturarse.

En Pruebas (Testing/Review): Fundamental en desarrollo (DAM). Antes de darla por buena, otro compañero o el tester verifica que el código funciona y no tiene *bugs*.

Terminado (Done): La tarea está completa, probada y lista para enseñarse al cliente en la *Sprint Review*.

8. Por si no te habías dado cuenta, antes te he engañado, no hemos comenzado todavía el desarrollo. Pero ahora sí que sí, tenemos nuestro panel Kanban y vamos a comenzar con las tareas de desarrollar. Vamos a ver que podría pasar durante el desarrollo.

Durante el Sprint no todo sale siempre según lo planeado. Al trabajar con el tablero Kanban pueden ocurrir distintas situaciones:

- Bloqueos: una tarea no puede avanzar por dependencia de otra, falta de información o un problema técnico.
- Cambios de prioridad: el Product Owner puede detectar que una tarea es más importante que otra y reajustar el orden.
- Errores o bugs: tareas que pasan a Testing/Review y vuelven a En Proceso porque fallan.
- Sobrecarga de trabajo: demasiadas tareas en Work In Progress (WIP), lo que ralentiza al equipo.
- Tareas no terminadas: algunas historias pueden no completarse dentro del Sprint y volver al Product Backlog.

Esto se arregla con Daily meetings, en los cuales se comenta:

- Qué hizo ayer
- Qué hará hoy
- Si tiene algún impedimento

9. Imaginemos que hemos terminado el primer Sprint, ¿qué toca ahora? Dos reuniones más:

- Revisión del Sprint:
- Retrospectiva del Sprint:

Revisión del Sprint (Sprint Review)

- Se presenta al cliente o Product Owner el trabajo terminado.
- Se muestran las funcionalidades completadas.
- Se recoge feedback para mejorar o ajustar requisitos.
- Se decide si las historias están realmente “Done”.

Retrospectiva del Sprint

Reunión interna del equipo.

Se analiza:

- Qué ha funcionado bien
- Qué ha ido mal
- Qué se puede mejorar en el próximo Sprint

Con el objetivo de mejorar la forma de trabajar, no el producto.

10. ¿Y ahora qué?

- Se actualiza el Product Backlog con nuevas historias o cambios.
- Se priorizan nuevamente las tareas.
- Comienza un nuevo Sprint, repitiendo el ciclo:
 1. Sprint Planning
 2. Desarrollo
 3. Revisión
 4. Retrospectiva