# La importancia de la priorización y las diferentes técnicas para priorizar

Históricamente, la **priorización** de los requisitos (software) siempre ha sido un tema crucial, ya fuera en la visión tradicional de los requisitos, o en su visión más ágil, donde lidiamos con historias de usuario y demás.

En el contexto de las metodologías ágiles, y más concretamente de Scrum, sabemos que el Product Backlog representa una lista de requisitos u objetivos priorizada, de acuerdo al valor que cada ítem de la lista aporta al cliente. Es el product owner, que hace las veces de cliente, el responsable de asignar (o averiguar) dicho valor y por tanto es él quien se encarga de priorizar el Product Backlog.

Pero, aunque tenemos claro que no debemos priorizar atendiendo al esfuerzo o la complejidad de desarrollar ese ítem no es lo adecuado, seguimos teniendo dudas en cuanto a cómo priorizar el Product Backlog. Por ello, en los últimos años han ido apareciendo diferentes técnicas de priorización de requisitos, ítems, historias de usuario, etc., en el contexto del mundo ágil.

## Filtro de priorización

En 2008, Corey Ladas (el autor de "Scrumban") difundió el **filtro de priorización**.

En su filtro de priorización, las columnas se etiquetan de izquierda a derecha como prioridad 3, prioridad 2 y prioridad 1. A su vez, cada columna tiene un WIP (te dejo aquí más información sobre el WIP, por si no conoces esto del WIP), es decir, un límite máximo de historias que puede haber en cada columna. Típicamente, la columna de prioridad 1 tiene un

límite de 1, es decir, sólo puede contener un requisito, ítem o historia de usuario.

Si quieres saber más sobre el tema, puedes leer una explicación más detallada aquí.

## Pirámide de Priorización

Es un derivado del anterior, de un tal Tomas Rybing, que aplicaba las mismas ideas pero proponía dibujar el product backlog en forma de pirámide en lugar de usar columnas.

De nuevo, puedes encontrar información detallada sobre esta forma de priorización aquí.

## MoSCoW y Kano

El modelo Kano, de los años 80, se utiliza para clasificar y priorizar requisitos en función de lo que satisfarán al usuario. Y para ello los divide en cuatro tipos: Requisitos obligatorios (básicos); Necesidades (esperado, lineal); No esperados (inesperado, excitante); e Indiferentes (el cliente no está interesado en ellos).

Por otro lado, el método **MoSCoW** hace algo similar, identificando otros cuatro tipos de requisitos: Must (obligatorios), Should (deberían estar), Could (podrían) y Won't (por ahora los dejamos).

Tanto el modelo Kano, como el MosSCoW se explican con más detalle en el libro "Gestión de Proyectos Ágiles …y las experiencias de más de 12 años de proyectos ágiles"