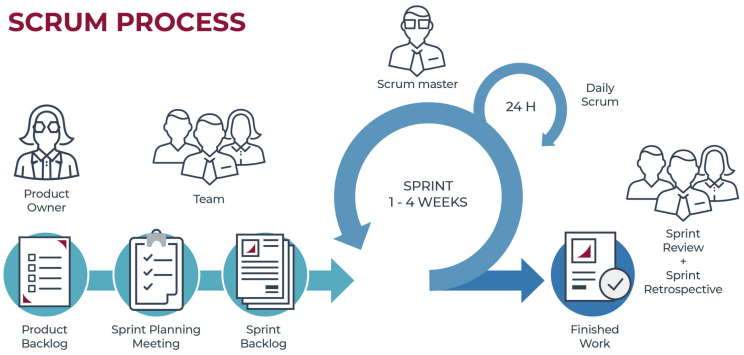
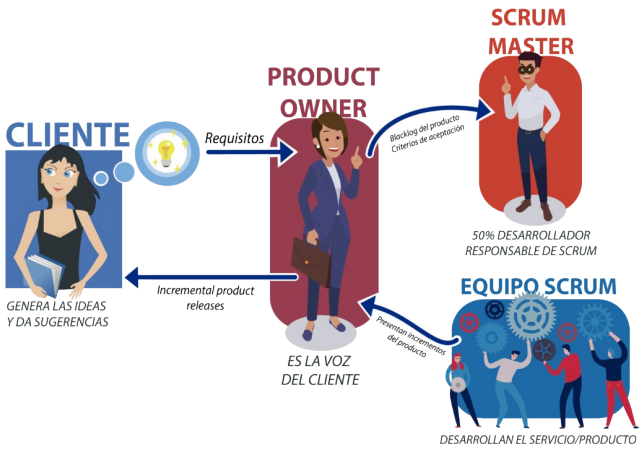
Implementación de Scrum

Scrum es un marco de trabajo ágil, con una serie de buenas prácticas y criterios para realizar un trabajo en equipo y obtener los mejores resultados, mejorando la eficiencia cada vez más mediante la la reflexión y la mejora continua.

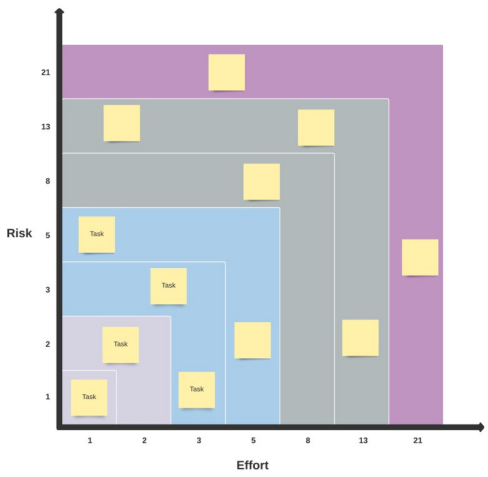


Para implementar Scrum se deben seguir una serie de pasos que se explican a continuación:

1. Elegir un responsable del producto: Es necesario seleccionar a una persona que posee en concreto la visión de lo que se quiere lograr o producir. Es quién debe percatarse de los posibles riesgos y de como actuar ante las posibles situaciones que lleguen a causar problemas para llegar al objetivo.
2. Selecciona un Equipo: Se debe tener un equipo multidisciplinario que sea capaz de trabajar en conjunto y debe tener al mismo tiempo las habilidades necesarias para llegar al objetivo a través de la visión de los responsables del producto. Por regla general, los equipos deben ser lo suficientemente pequeños para mantener la agilidad y lo suficientemente grandes para lograr el trabajo necesario, regularmente de 3 a 9 personas.
3. Elige un Scrum Master: El Scrum Master es el individuo que se encargará de capacitar al equipo en el enfoque del marco de trabajo y le permitirá ver al equipo las situaciones que atrasan el trabajo o que lo llevan por un camino equivocado para poder realizar las correcciones adecuadas.



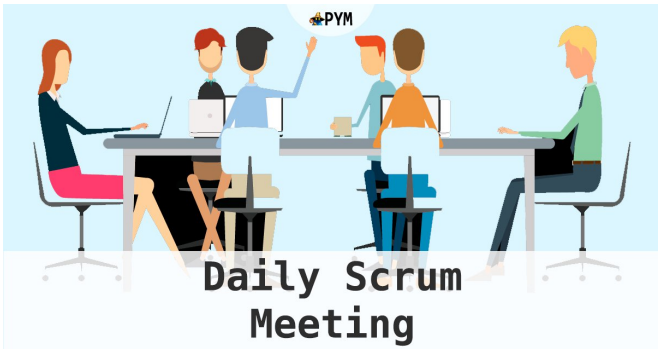
4. Crea y prioriza una bitácora del producto: Llevar una bitácora con las tareas a realizar y con lo necesario para cumplirlas es de gran ayuda, ya que permite tener ideas más claras y formar una visión más solida. Lo que se escriba en la bitácora no es estático y puede cambiar y evolucionar en el periodo de vida del producto. El responsable del producto es quien debe tomar las decisiones de priorización sobre las tareas, esto mediante reuniones y acuerdos entre el equipo de trabajo que permitan esclarecer las actividades a llevar a cabo y su importancia.
5. Afina y estima la bitácora del producto: Teniendo una bitácora con las tareas y sus prioridades, corresponde que la gente que se hará cargo verifique que las tareas son viables y realistas. Es importante esclarecer una forma adecuada de medir la forma en que una tarea es cumplida o terminada. Se recomienda no calcular la bitácora en horas, sino mediante un valor relativo (pequeño, mediano o grande), o mediante la serie Fibonacci con un valor puntual de cada elemento.
6. Planificación del sprint: La primera reunión de Scrum, donde el Scrum Master y el responsable del producto planean el sprint. Los sprint son de extensión fija en un lapso de tiempo menor a un mes. Se suelen realizar uno o dos sprints semanales. El equipo debe examinar el inicio de la bitácora y pronostica cuanto puede llevar a cabo en este sprint. El equipo debe considerar al número de puntos acumulados en cada sprint, conocido como velocidad del equipo, del cual se debe buscar que cada vez sea mayor por cada sprint. Una vez que el equipo se compromete con lo que cree que es capaz de terminar en un sprint, se debe mantener en esto, no puede cambiar ni tampoco crecer.
7. Vuelve visible el trabajo: Las personas suelen preferir los elementos visuales y son rápidamente más comprensibles, por lo que una buena manera de demostrar el avance es mediante una tabla con 3 columnas: Pendiente, En proceso y Terminado. Otra forma es mediante un diagrama de finalización, con el número de puntos que el equipo introdujo en el sprint y en otro el numero de días, sumando cada día el numero de puntos completados.



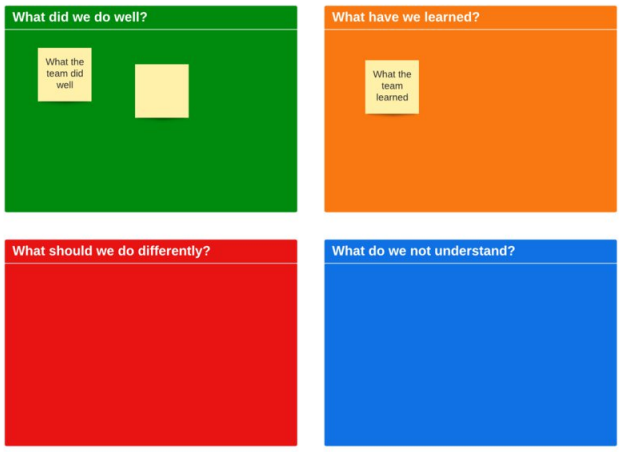
8. Parada diaria o Scrum diario: El equipo Scrum debe de realizar reuniones diarias para asegurarse que todos se han alineado. Cada día, a la misma hora, durante no más de quince minutos, el equipo debe reunirse y contestar las preguntas:

a. ¿Qué hiciste ayer para ayudar al equipo a terminar el sprint?
b. ¿Qué harás hoy para ayudar al equipo a terminar el sprint?
c. ¿Algún obstáculo te impide o impide al equipo cumplir la meta del sprint?

Esto permite ayudar al equipo a saber dónde se encuentra actualmente y que se necesita hacer para llegar al objetivo.



9. Revisión del sprint o demostración del sprint: Al final del sprint, el equipo debe someterse a una revisión para evaluar qué se logró y lo que se puede mejorar en el futuro. Pueden asistir todos, incluidos el responsable del producto, el Scrum Master, el equipo, la dirección, clientes y demás interesados. Se debe presentar y entregar únicamente lo que se ha Terminado completamente.
10. Retrospectiva del sprint: El equipo de Scrum continuamente está mejorando su eficiencia en cada sprint. Se debe pensar y reflexionar sobre qué marchó bien, qué pudo haber mejorado y qué puede mejorar en el siguiente sprint. Es crucial un buen manejo emocional, una atmósfera de confianza y que el equipo sea capaz de asumir su responsabilidad sobre los resultados obtenidos. Al final de la reunión el Scrum Master debe formular una mejora al proceso llamada “kaizen” que se tiene que incorporar a la bitácora del sprint.



11. Comienza de inmediato el ciclo del sigueinte sprint, tomando en cuenta la experiencia del equipo con los impedimentos y mejoras del proceso.

Git (Branches, Merge y Conflicts)

Branches

Un branch (o rama) es una línea de desarrollo separada en un repositorio git. Es posible crear nuevas branches en Git para desarrollar características, solucionar un error, experimentar con ideas sin afectar el código principal, etc.

Para crear una nueva rama, primero se debe estar en la rama principal o en aquella de la que se desea derivar la nueva rama. Los comandos a utilizar son:

```
//Mover a la rama principal
$ git checkout main

//Crear nueva rama
$ git branch nueva-rama

//Los comandos anteriores creara una nueva rama a partir de la rama master

      o---o---o---o---o master
          \
           o---o---o nueva-rama
```

Se puede usar el comando git branch para ver una lista de todas las ramas que existen en el repositorio.

```
//Mostrar todas las ramas disponibles en el repositorio
$ git branch

  main
* nueva-rama
```

Merge

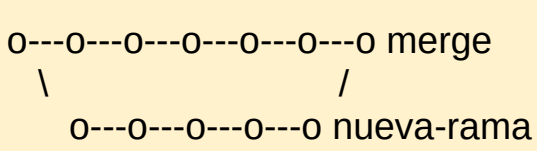
Una vez que se ha desarrollado una nueva característica en una rama separada, será necesario unir los cambios con el branch principal (main) para incorporarlos en el proyecto. Esto se conoce como merge (fusionar).

Antes de realizar este procedimiento primero se debe asegurar estar en la rama a la que se necesita unir los cambios, para luego utilizar el comando git merge con el nombre de la rama que contiene los cambios a unir.

```
//Mover a la rama principal -> Rama a la que se desea añadir los cambios
$ git checkout main

//Realizar merge -> Rama con los cambios que se desean aplicar
$ git merge nueva-rama

//La rama main ahora contiene los cambios realizados en nueva-rama
```



Este comando fusionará los cambios realizados en el branch "nueva-rama" en la rama "main".

Conflicts

Si Git encuentra conflictos entre las ramas que estás tratando de fusionar, solicitará que se resuelvan los conflictos manualmente antes de poder continuar con la fusión. Esto se debe a que Git no sabe cómo combinar los cambios es necesario decirle cómo hacerlo.

Estos conflictos (en inglés Conflicts) ocurren cuando dos ramas han realizado cambios en las mismas líneas del mismo archivo. Cuando Git encuentra un conflicto, mostrará un mensaje similar a este:

```
Auto-merging archivo.txt
CONFLICT (content): Merge conflict in archivo.txt
```

Para solucionar el conflicto, se debe editar manualmente el archivo con conflicto y seleccionar qué cambios se desean conservar. El archivo en conflicto puede aparecer con una estructura similar a como se ve a continuación:

```
<<<<<<< HEAD
Cambios que tú hiciste.
=======
Cambios que otra persona hizo.
>>>>>>> nueva-rama
```

Se debe elegir que cambios son los que se desean conservar, y una vez editado el archivo conflictivo, se debe confirmar a git la resolución del conflicto con git add y git commit.

```
$ git add archivo.txt
```

```
$ git commit -m "resuelvo conflicto de merge"

//Al fusionarse con conflictos se debe arreglar los conflictos de forma manual

          o---o---o---o---o---o main
              \               /
                o---o---X---o nueva-rama

//Al realizarse un commit después de solucionar los conflictos, las ramas
//se fusionarán sin problemas

          o---o---o---o---o---o---o main
              \               /     /
                o---o---X---o nueva-rama
```