

## Habilidades blandas para la gestión

Gestionar un proceso de software no es para nada una tarea trivial, ya que de esto podría depender en mayor o menor medida el éxito del mismo. Es por eso que se necesitan de ciertas habilidades con las cuales se determinarían una serie de tareas a realizar y así lograr un objetivo determinado, que en este caso es el desarrollo de software.

Existen muchas habilidades blandas que podrían ayudar a gestionar un proceso de software, pero a mi consideración hay 3 que son vitales: gestión del tiempo, organización y pensamiento crítico.

**Gestión del tiempo:** la gestión del tiempo es uno de los principales pilares de la productividad y hace referencia a la planificación y control del tiempo que se dedica a determinadas tareas. En un proyecto de software podemos encontrar el “triángulo de hierro”, en el cual se debe de tomar en cuenta el alcance, el tiempo y el dinero para poder tener un balance y completar los objetivos establecidos.

El tiempo no se puede comprar, es por eso que debemos saber cómo gestionar nuestras horas, priorizando las actividades más relevantes y evitando distracciones.

**Organización:** otra habilidad muy importante para la gestión de proyectos es la organización, la cual busca tener una estructura y un orden establecido en ciertas tareas, priorizando la productividad y las tareas que deben completarse inmediatamente para así poder mostrar avances a los interesados.

**Pensamiento crítico:** con esta habilidad podemos tener la capacidad de analizar y evaluar ciertos aspectos indispensables para gestionar correctamente un proceso de software. Gracias al pensamiento crítico podemos utilizarlo para llegar a la forma más objetiva de las decisiones tomadas.

## **Sprint**

Un sprint es un período breve de tiempo fijo en el que el equipo de scrum trabaja para completar una cantidad de trabajo establecida. En mi equipo, los sprints los planificábamos de acuerdo a las fechas de entrega y lo que necesitábamos entregar, es decir, todas las actividades necesarias para poder completar los requerimientos establecidos.

Considero que hicimos un buen trabajo tomando como marco de referencia Scrum, pero siempre hay cosas que podemos mejorar. En mi experiencia lo que cambiaría podría ser tener una mejor organización al distribuir las actividades por hacer, ya que solamente las repartíamos conforme a lo que necesitábamos completar y nunca tomamos en cuenta una estructura específica o algo de ese estilo.

Para futuros proyectos para tener mejores resultados lo que tomaría principal en cuenta es la organización del equipo y siempre poniendo fechas límite para así evitar atrasos en el proceso.

## **Calidad Ágil**

Existen diversos modelos para asegurar la calidad del software, los más conocidos son CMM y PSP los cuales explicare a continuación.

CMM ,que por sus siglas en ingles significa “modelo de capacidad y madurez”, tiene como objetivo evaluar los procesos de software en sus distintos niveles de madurez, identificar los niveles a través de los cuales una organización debe formarse para establecer una cultura de excelencia en la ingeniería de software.

PSP que sus siglas quiere decir “proceso personal de software”, es una tecnología que tiene como justificación la premisa de la calidad de software depende del trabajo de cada uno de los ingenieros de software y de aquí que el proceso diseñado debe ayudar a controlar, manejar y mejorar el trabajo de los ingenieros.

En Scrum la calidad es primordial, la calidad se define como la capacidad que tienen los entregables o el producto terminado de cumplir con los criterios de aceptación y lograr el valor del negocio que espera el cliente.

## **Tema libre: Creatividad en ciencias computacionales**

La creatividad es la habilidad de desarrollar nuevas ideas para resolver problemas y comunicarse con los demás. Esas personas que son creativas pueden poner sus ideas en marcha, es decir, traerlas a la realidad.

La creatividad es importante ya que nos ayuda a resolver problemas. Nos brinda confianza al entrar en situaciones nuevas, sabiendo que puedes solucionar problemas. La creatividad nos anima a aprender e interactuar con otros. También en cuanto más conocimiento compartimos entre nosotros, más podemos aprovechar para desarrollar nuevas ideas.

Si bien la mayoría de las personas no asocian inmediatamente la creatividad con la informática, las habilidades creativas no solo son beneficiosas para los desarrolladores, sino que a menudo son críticas durante el proceso de desarrollo. Cuando los desarrolladores comienzan a codificar un programa, definen el problema e imaginan posibles soluciones. Los informáticos se benefician de un sinnúmero de posibilidades y caminos de solución, donde cualquier solución es válida. Aquí es donde entran en juego la resolución creativa de problemas y la originalidad.

Experimentan y ponen en práctica sus pensamientos e ideas utilizando el código como medio, al igual que los escritores comienzan poniendo la pluma sobre el papel para crear algo nuevo. Luego, iteran y depuran hasta que descubren una solución que funciona. Una vez que tienen un programa viable, iteran un poco más para refinar su creación, como hacer que el código sea más conciso o mejorar la legibilidad.

De manera similar a como un pintor puede afinar continuamente una obra de arte o un escritor puede revisar un artículo muchas veces, un desarrollador siempre está reelaborando su código y desarrollando su propio estilo. Aprender a codificar requiere que los desarrolladores sean creativos, especialmente porque pueden estar resolviendo problemas que no se han resuelto antes.

Sandy Dibble. 2021. Creativity in computer science. In Codelicious. Retrieved November 30, 2022 from

<https://www.codelicious.com/blog/creativity-in-computer-science#:~:text=While%20most%20people%20do%20not,problem%20and%20imagine%20potential%20solutions.>