

METODOLOGIAS TRADICIONALES VS METODOLOGIAS AGILES

Al principio todo el proceso del software era de manera artesanal, esto es porque las necesidades de cada quien los llevaban a cambiar o modificar algún software para cumplir una meta a la cual querían llegar. Dicho proceso se dividía en varias etapas para mejorar el avance del software.

Entre las metodologías tradicionales destacan dos RUP y MSF que se encargan de hacer la documentación de todo el software, con su plan de proyecto y hasta donde quiere que el proyecto llegue esto es de lo que se encargan las dos principales metodologías tradicionales.

Tienen una característica importante que son los altos costos de implementación y esto es por pocas soluciones al problema encontrado relacionado al proyecto del software. Las metodologías tradicionales se enfocan en la documentación planificación, y procesos. Esto es lo que se hace llevando en ellas el control de todas las modificaciones hechas, así como de igual manera las revisiones y opiniones.

Rational Unified Process (RUP)

Es un proceso que prevé todas las actividades a realizar en el proyecto asignando así mismo como planificar las tareas, así como el desarrollo dentro de la organización. Su objetivo final es dar un buen servicio y un buen software que satisfaga las necesidades del usuario pero que este sea respetando las tareas y actividades a realizar en el tiempo acordado para así dar un mejor rendimiento a la hora de producir el software.

Fue desarrollado por relational software e integrado por la suite relational de herramientas.

Este proceso puede ser adaptado por cualquier empresa para poder realizar muy bien su trabajo por lo que la metodología realiza, y que cada empresa lo puede acoplar a su necesidad y forma. Para poder darle un buen uso es más fácil hacerlo por medio de diagramas de caso de uso y diagramas UML.

La RUP tiene como fases la:

Concepción, elaboración, construcción y transición

VENTAJAS RUP:

- Cada fase del proyecto debe de ser evaluada y autorizada para poder realizar cambios.

- Por su proceso ayuda más en proyectos de innovación
- Es de uso fácil ya que se basa en los diagramas con procesos ya especificados, para la realización del software.
- Cada paso a realizar lleva cierto detenimiento para hacerlo con mayor facilidad.

DESVENTAJAS URP:

- Cada evaluación de riesgos lleva un proceso.
- Cualquiera la puede usar de manera que quiera y en el proyecto que quiera
- Pone al usuario de manera muy importante porque es el que dice cómo y cuando requiere de su proyecto.
- Cada cliente debe de tener bien en claro como es el proyecto para que pueda describir la realización de su proyecto.

Microsoft **S**olution **F**ramework (**MSF**)

Se puede adaptar a cualquier proyecto de tecnologías de la información, es una metodología que se encarga mucho más de la administración de proyectos. Se divide en 5 fases que son: Visión y alcances, planificación, desarrollo, estabilización e implementación.

Extreme **P**rogramming (**XP**)

Es la que destaca más en las metodologías ágiles, se considera que cualquier cambio es de manera natural puesto que no afecta a la metodología, también dicen que todo proyecto debe estar apto para los cambios que se le hagan sin ningún problema.

Características de la metodología:

- **Desarrollo iterativo e incremental:** mejoras y cambios en el proceso unas con otras.
- **Pruebas unitarias:** continuamente se repiten. Aunque antes de hacer varias repeticiones escribir el código primero antes de cada repetición con las mejoras ya hechas.
- **Programación por parejas:** el trabajo realizado este mejor hecho por dos personas que una, porque el trabajo uno lo realiza y el otro verifica los errores para corregirlos y tener más avances.

- **Frecuente interacción:** tiene mucha comunicación con el cliente para verificar si el trabajo realizado va cumpliendo con las expectativas del cliente y los requerimientos ya antes pedidos.
- **Corrección:** después de cada error es mejor corregir y después entregarlo de manera correcta con las mejoras ya realizadas.
- **Refactorización:** es cuando se escribe el código nuevamente, no completamente pero tiene algunas partes que es necesario hacerlo para mayor legibilidad y autenticidad.
- **Propiedad del código compartida:** permite que todo el equipo de trabajo pueda participar en la elaboración, corrección e incluso pueden tener ideas de mejoramiento del sistema y tienen que aprobar la idea los demás miembros del equipo, la participación en el son de manera equitativa.
- **Simplicidad en el código:** es la manera más fácil de realizar el trabajo, y a la hora de querer hacer modificaciones ya con el código un poco más explícito este podrá ser modificado ya sea agregando o quitando recursos del software.

Ventajas:

- Reduce costes por estar aptos para cada cambio o modificación.
- Cuando un cliente pide un sistema es más fácil la entrega en la fecha acordada puesto que el sistema puede ser basado en algún otro sistema antes realizado aplicando algunas modificaciones.
- Permite tener mas pronto un prototipo para recibir retroalimentación por parte del cliente (usuario), basándose en ella para la entrega en forma del sistema.
- Siempre tendrás que mostrar los avances del proyecto, así evitando retrasos por falta de retroalimentación.

Desventajas:

- Se delimita solo a realizar lo que el cliente pide.