

# Introducción a las Metodologías de Desarrollo de Software:

## 1 - ¿Qué es una metodología de desarrollo de software?

Una metodología de desarrollo de software no es mas que una serie de pasos que se realizan de forma rigurosa tal que su resultado a partir de unos requisitos nuevos o modificados sea un software nuevo o modificado. Se puede ver como una caja negra, como muestra la siguiente imagen:



Esta nos permite responder a cuatro preguntas básicas:

- ¿Qué hacen los participantes del proyecto?
- ¿Quién participa en su desarrollo?
- ¿Cómo participan los anteriores?
- ¿Cuándo se realizan sus partes?

Personas realizan actividades según sus roles a partir de resultados parciales (conocidos como artefactos) que son partes necesarias para realizarlas o resultados parciales o totales de éstas. Estas actividades toman un orden y una estructura temporal entrelazada definida por la metodología en cuestión.

## 2 - ¿Qué nos aporta una metodología de desarrollo de software?

La gracia de una metodología es que aporta una garantía de calidad.

¡IMPORTANTE!

Un producto de software es de calidad si cumple rigurosamente con todos y cada uno de sus requisitos. Es decir, calidad = requisitos satisfechos.

Gracias a esto podemos medir la calidad de un producto basándonos en los requisitos iniciales.

También nos aporta una forma de estimar y controlar costes. Así podemos saber cuanto vamos a tardar en realizarlo y si nos sale o no rentable llevarlo a cabo antes de realizar la inversión completa de tiempo, dinero y esfuerzo. También evita una gran parte de los esfuerzos perdidos en rectificar fallos que se pueden evitar utilizando una metodología adecuada.

Al ser un proceso estructurado también nos organiza la forma en la que el proyecto va a ser realizado, obligando a revisar que los resultados sean los correctos antes de proseguir y marcando metas intermedias para controlar el avance del proyecto. Así pues, se logra una mayor eficiencia de recursos, es decir, se invierte lo mínimo para obtener lo máximo a cambio. Para que el proceso sea efectivo, éste debe ser aplicado con rigor.

### 3 – ¿Qué metodología escoger?

Existen dos tipos principales de metodologías, las Ligeras y las Pesadas. Las primeras son metodologías extremadamente prácticas que generalmente obvian gran parte de la documentación y están mas preparadas para utilizarse en proyectos cuyos requisitos cambiarán constantemente durante todo el proceso.

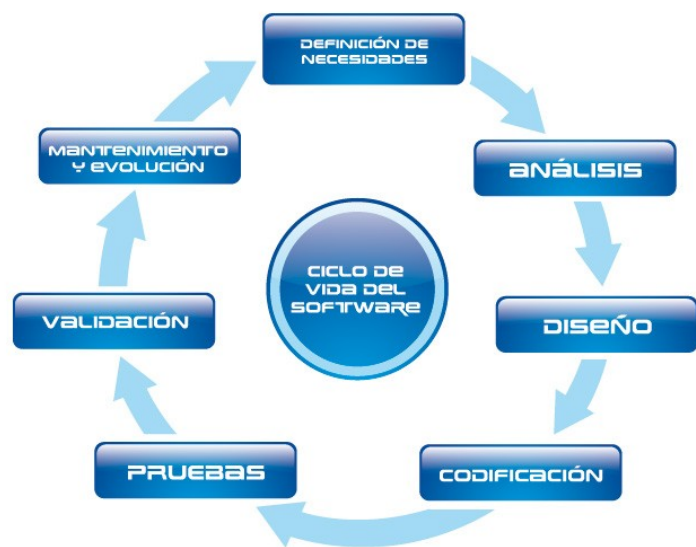
Las segundas, son metodologías donde todo está mucho más controlado y se genera muchísima documentación antes de proceder a implementar el proyecto, con mucho mayor peso del análisis y el diseño sobre el proyecto. Estas últimas son más indicadas para proyectos grandes o cuyo rendimiento y nivel de calidad son críticos para el éxito de éste.

Ejemplos de metodologías ligeras podrían ser eXtreme Programming (XP), SCRUM y crystal, mientras que ejemplos de metodologías pesadas podrían ser Rational Unified Process (RUP), ICONIX y Métrica 3.

### 4 - ¿Y qué hay del ciclo de vida del software?

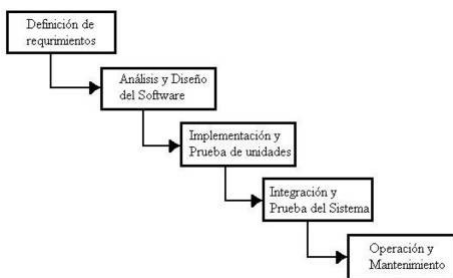
El ciclo de vida del software es el conjunto de etapas que sigue un proyecto de software desde su concepción hasta su finalización y cierre, inclusive los mantenimientos (es decir, cambios o ajustes que puedan producirse una vez está implementado, nuevas versiones, etc.).

Aquí se puede observar un ejemplo de ciclo de vida del software. Éste se inicia con la definición de necesidades y sigue un flujo cíclico hasta retornar al punto de origen.



A continuación mostraremos diversos enfoques que puede tomar una metodología de desarrollo de software y explicaremos las diferencias fundamentales entre ellas.

Estos enfoques serán: Cascada, Prototipaje, Incremental y Espiral.



#### - Cascada:

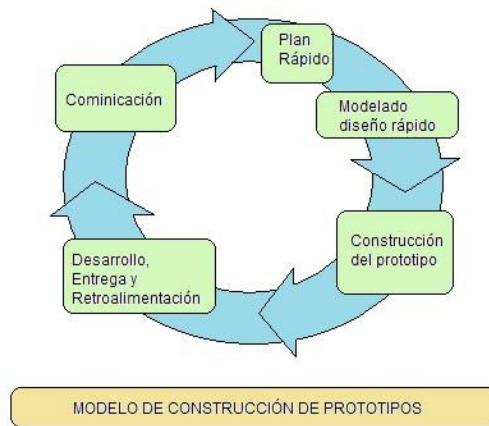
Como se puede observar, se trata de un enfoque secuencial.

En este caso, cualquier fallo de las fases anteriores serán arreglados en la fase actual, y se procederá siempre hacia adelante, sin volver a pasar por ninguna de las fases anteriores.

## - Prototipaje:

En este caso el enfoque es iterativo.

Se basa en realizar pequeños prototipos finales de la aplicación de forma que sus funcionalidades se construyen encima de la versión anterior, hasta llegar al producto definitivo y su entrega al cliente.



## - Incremental:

Nótese que esta vez el enfoque es combinado.

Este caso es similar al del prototipaje, pero lo que ocurre es que se van haciendo mini-cascadas en cada iteración, de forma que pasa por todas sus fases.

Una vez acabada una mini-cascada, comienza la siguiente iteración, y así sucesivamente.

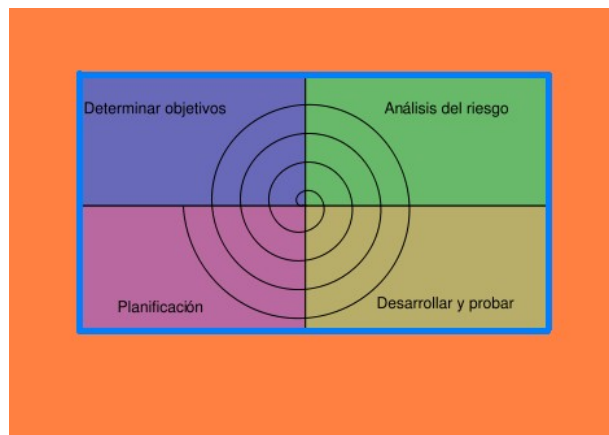


## - Espiral:

Se trata de otro enfoque combinado, pero mucho más complejo que los anteriores.

Se puede observar el proceso como una espiral. Cada rotación representa una mini-cascada, y la distancia radial representa el volumen del proyecto.

A más avanzado, mayor volumen.



Lo que ocurre con esta metodología es que su coste es bastante impredecible debido al volumen del proyecto, con lo cual no suele ser un enfoque viable económicamente. Sin embargo, tiene sus usos cuando se realizan proyectos críticos como un gran sistema operativo, temas de control aéreo, militares o espaciales, ya que prima la calidad sobre el coste principalmente. Un solo fallo puede ser motivo de su completo fracaso.

## 5 - Referencia Recomendada: Metodologías de Desarrollo del Software.

<https://pid.dsic.upv.es/C1/Material/Documentos%20Disponibles/Introducci%C3%B3n%20Proceso%20de%20Desarrollo%20de%20SW.doc>