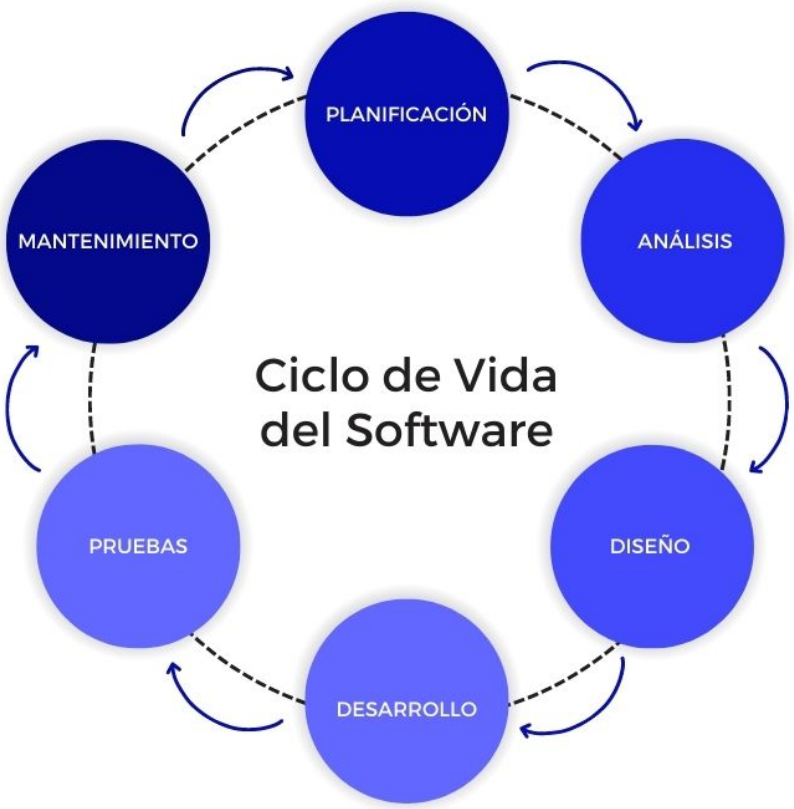
Ciclo de Vida del Software

evotic.es/software-a-medida/ciclo-de-vida-del-software

Etapas y modelos del Ciclo de Vida de un Software

El **ciclo de vida del software** nos permite identificar, administrar y planificar la gestión de recursos hasta alcanzar un objetivo propuesto. La cantidad de fases que encontramos en cada proyecto variará según las necesidades de cada empresa.



DESARROLLO SOFTWARE

Conocer el ciclo de vida de un proyecto permite **controlar mejor sus fases** y reducir situaciones innecesarias. En cualquier proyecto, las fases son las mismas, aunque cada proyecto requiere su propio conjunto único de procesos. Normalmente, encontramos una fase inicial, un proceso de desarrollo y un producto final. A este patrón se le denomina ciclo de vida de un software.

En este artículo queremos ayudarte a entender el ciclo de vida de un proyecto, analizaremos fase por fase, todas sus etapas y modelos. ¡Sigue leyendo!

Etapas del ciclo de vida del software

Para entender mejor la evolución en el ciclo de vida de un proyecto vamos a analizar todas sus etapas, esto nos ayudará a planificar una hoja de ruta con la que concluir satisfactoriamente todos los procesos.

1- Fase de Planificación

Esta será una de las primeras fases, en la cual, se trazan objetivos y se valoran las necesidades del proyecto en cuestión. Las tareas iniciales consisten en realizar un estudio de viabilidad, analizar los riesgos que puede comportar el proyecto y planificar todas las fases posteriores. A continuación, se documentan en detalle las funcionalidades, características y objetivos que debe cumplir el ciclo de vida del proyecto.

- El **estudio de viabilidad** nos permite evaluar los aspectos técnicos, financieros y legales, para poder determinar si el proyecto podrá ser completado con los recursos disponibles.
- **Analizar los riesgos** nos puede ayudar a estar preparados para cualquier contratiempo.
- **Planificar el proyecto** fase por fase nos permite llevar a cabo su desarrollo con éxito.

2- Fase de Análisis

En la fase de análisis de un proyecto definimos **que funciones ejecutará el software** y cuáles son sus características específicas. Este aspecto es clave a la hora de optimizar la asignación de costes y determinar la envergadura del proyecto.

La **asignación de costes** es una de las tareas más complejas dentro de un proyecto de desarrollo de software, ya que se debe estimar al inicio, cuando menos conocemos el proyecto y mayor es el margen de error. Afortunadamente, la experiencia en proyectos similares y la fragmentación del proyecto por tareas nos facilita mucho el hecho de presupuestar adecuadamente.

3- Diseño y estructura del software

Una vez que disponemos de toda la información que precisa nuestro proyecto, estudiaremos las posibles alternativas y estableceremos la estructura de la base de datos, la lógica del flujo de datos y la interfaz de usuario de la aplicación.

- **Estructura de la base de datos.** Este proceso es clave en el ciclo de vida de un software. Tomar decisiones adecuadas facilitará la implementación del proyecto. El acceso a los datos almacenados permitirá al usuario poder gestionar la información que precisa.
- **Lógica del flujo de datos.** Crear un diagrama de flujo de datos, nos permite trazar el flujo de información de cualquier proceso o sistema. Estos diagramas nos dan información detallada de nuestro proyecto de forma visual.
- **Interfaz de usuario.** Diseñar un entorno intuitivo y fácil de usar, es otro elemento clave para un proyecto de desarrollo de software. Que el usuario sea capaz de acceder a los datos y gestionar el aplicativo de manera ágil, es un requisito indispensable.

4- Fase de Desarrollo de un Software

Cuando ya hemos definido la estructura de nuestro software, llega el momento de empezar con la programación. Escoger el **lenguaje de programación adecuado** a nuestro proyecto y disponer de un equipo de programadores expertos es fundamental. Seguir el código de buenas prácticas permitirá que nuestro proyecto sea fácilmente escalable.

5- Fase de Pruebas

Una vez terminado el proceso de desarrollo empieza el testeo y la fase de pruebas de la aplicación. En esta etapa ponemos a prueba los errores que hayan podido aparecer en las etapas anteriores. Es una fase de corrección, eliminación y perfeccionamiento de posibles fallos, no previsto en los pasos previos.

6- Fase de Mantenimiento

En este periodo el software ya está en funcionamiento. Con el tiempo alguna función puede quedar obsoleta, pueden detectarse algunas limitaciones o que aparezcan propuestas que mejoren la estabilidad del proyecto.

Modelo de ciclo de vida de un software

Los modelos nos permiten establecer una metodología que nos servirá de guía para validar todas las partes de nuestro proyecto. A continuación, vamos a ver cuáles son los distintos modelos que se utilizan en el desarrollo y la preparación de un software.

1- Modelo en cascada

El modelo en cascada es un método de gestión que divide un proyecto en distintas fases secuenciales. Todas ellas funcionan de forma lineal, es decir, que cada parte del proyecto se completa antes de empezar con la siguiente. Este modelo es simple y fácil de usar, ya que todo está bien organizado y las fases no pueden mezclarse entre sí.

2- Modelo en V

El modelo en V está inspirado en el método cascada. Se basa en la gestión de proyectos de forma lineal, pero consta de una fase descendente y otra ascendente.

- **Fase descendente:** Tareas de diseño y desarrollo.
- **Fase ascendente:** Control de calidad.

3- Modelo de prototipos

El modelo de prototipo permite al usuario obtener una visión previa de las principales características del software. Debe ser construido en poco tiempo y la aplicación representará los aspectos del proyecto que serán visibles para el cliente final.

Una vez testeada por los usuarios, aparecerán necesidades que se habían pasado por alto y esto conlleva a la realización de nuevas versiones hasta que se desarrolle el sistema o producto final.

4- Modelo en espiral

El modelo de desarrollo en espiral se organiza en distintas fases. La espiral conforma un conjunto de actividades, en la que cada bucle o iteración representa un conjunto. En un primer momento, se definen los objetivos con el fin de empezar con la fase de desarrollo. A continuación, se evalúan las posibles mejoras, finalizando con la fase de mantenimiento.

5- Modelo de desarrollo incremental

El modelo incremental se diferencia por dividir sus tareas en iteraciones, trabajando objetivos en pequeños tramos específicos. Estas iteraciones permiten una fácil administración y están sujetas a cambios y modificaciones según las necesidades que puedan ir surgiendo.

Las **herramientas más conocidas** que utilizan este tipo de modelo de **gestión de proyectos** de software son Scrum y Kanban, también llamadas **Metodologías Ágiles.**

Métodos Ágiles

Los métodos ágiles plantean métodos de gestión de proyectos menos rígidos, donde la interacción del cliente es clave. Estos métodos mejoran enormemente la productividad y adquieren las siguientes características:

- **Entregas frecuentes y revisión del trabajo por pasos.**
- **Adaptación a posibles cambios tras recibir la interacción del cliente.**
- **Organización de reuniones, donde se plantean avances y se hace balance de los resultados.**
- **Mejoras del proyecto de forma continua.**

1- Scrum

El Método Scrum se caracteriza en trabajar por ciclos o Sprints. Es una metodología que exige equipos multidisciplinares, donde cada miembro del equipo puede cumplir varias tareas. Este método consta de una jerarquía de roles, el equipo se reúne diariamente con el fin de aunar criterios y realizar las adaptaciones necesarias.

2- Kanban

En el método Kanban no existen roles de jerarquía, cada miembro del equipo cumple con sus tareas concretas. No se trabaja por Sprints, ya que el flujo de trabajo es continuo. En Kanban los equipos están formados por especialistas, eso permite maximizar la eficiencia del proyecto.