

Metodología

Ciclo de Vida:

Es la forma en la que se denomina al conjunto de **tareas a realizar** (agrupadas en **etapas**) establecidas por la **metodología estándar de trabajo de un desarrollador de software**.

Etapas:

Obtención de los requisitos

Análisis de los requisitos:

Proceso inicial del ciclo de vida de un proyecto en el cual se obtienen los requisitos necesarios para armar el proyecto mediante conversaciones, ya sea con el cliente como con expertos funcionales en pos de armar un producto útil para los usuarios finales.

Se ven plasmadas en Diagramas de Entidad-Relación o documentos ERS (especificación de requisitos del sistema); los cuales conforman el diseño conceptual del proyecto.

Finalidades

- Brindar al usuario todo lo necesario para que pueda trabajar en conjunto con el *software* desarrollado obteniendo los mejores resultados posibles.
- Tener un control más completo en la etapa creación del *software*, en cuanto a tiempo de desarrollo y costos.
- Utilización de métodos más eficientes que permitan el mejor aprovechamiento del *software* según sea la finalidad de uso del mismo.
- Aumentar la calidad del *software* desarrollado al disminuir los riesgos de mal funcionamiento.

Limitaciones:

El software es diseñado con funciones **específicas** en mente, por lo cual no es capaz de reemplazar totalmente el pensamiento humano. Por ende, se debe de pensar en su utilidad como una **herramienta complementaria** para los usuarios.

Especificación:

La **especificación de requisitos** describe el comportamiento esperado del **software desarrollado**. Se pueden armar de dos formas:

- Casos de uso: Rigurosos y formales
- Historias de Usuario: Ágiles e informales

Arquitectura:

La arquitectura de *software* define **cómo se armará la aplicación**. Consiste entonces en el diseño de **componentes** de una aplicación (entidades del negocio), generalmente utilizando **patrones de arquitectura**. El diseño arquitectónico debe permitir **visualizar la interacción** entre las entidades del negocio y además poder ser validado, por ejemplo por medio de **diagramas de secuencia**:

- Diagrama de clases
- Diagrama de base de datos
- Diagrama de despliegue
- Diagrama de secuencia

Programación:

Implementación de los diagramas diseñados en código.

Desarrollo de la Aplicación:

Consiste en **5 fases importantes**:

- **Desarrollo de la infraestructura**
- **Instalación y preparación de las dependencias del proyecto (Adaptar Paquetes)**
- **Desarrollo de unidades de diseño de interactivas (UI)**
- **Desarrollo de unidades de diseño batch (flujo de la aplicación)**
- **Desarrollo de unidades de diseño manuales**

Pruebas de Software:

Testing del software, producible mediante **pruebas unitarias** (una función a la vez) e **pruebas de integración** (todo a la vez). Preferible tener un equipo designado (QA).

Implementación:

Proceso en el cual se transforma una especificación de un sistema en un sistema ejecutable.

Documentación

Mantenimiento

Ventajas de este sistema:

- **Desde el punto de vista de gestión**
 - Facilitar la tarea de seguimiento del proyecto
 - Optimizar el uso de recursos
 - Facilitar la comunicación entre usuarios y desarrolladores
 - Facilitar la evaluación de resultados y cumplimiento de objetivos
- **Desde el punto de vista de los ingenieros de *software***
 - Ayudar a comprender el problema
 - Permitir la reutilización
 - Facilitar el mantenimiento del producto final
 - Optimizar el conjunto y cada una de las fases del proceso de desarrollo
- **Desde el punto de vista de cliente o usuario final**
 - Garantizar el nivel de calidad del producto final
 - Obtener el ciclo de vida adecuado para el proyecto
 - Confianza en los plazos del tiempo mostrados en la definición del proyecto