## Modelo en cascada

### Introducción

El modelo en cascada, también conocido como el modelo de desarrollo de software en cascada, es uno de los enfoques más antiguos y tradicionales para el desarrollo de software. Fue propuesto por primera vez por Winston W. Royce en 1970. Este modelo es conocido como "cascada" debido a su enfoque secuencial y lineal para el desarrollo de software, donde el flujo de las fases se asemeja a una cascada.

El modelo en cascada se basa en la premisa de que cada fase del ciclo de vida del desarrollo de software debe completarse antes de que pueda comenzar la siguiente. Las fases típicas del modelo en cascada incluyen: requisitos, diseño, implementación, verificación y mantenimiento. Cada una de estas fases se considera un paso distinto y separado en el proceso de desarrollo de software.

### Descripción

El modelo de cascada, también conocido como el modelo de desarrollo lineal secuencial, es uno de los métodos más antiguos y tradicionales para el desarrollo de software. Este modelo sigue una secuencia lineal de fases que comienza con la definición de los requisitos del sistema y concluye con la entrega y el mantenimiento del producto finalizado. Cada fase del modelo de cascada debe completarse antes de que pueda comenzar la siguiente.

Este modelo es simple y fácil de entender y usar, lo que lo hace adecuado para proyectos pequeños donde los requisitos son muy bien conocidos, claros y fijos. Sin embargo, su rigidez puede ser una desventaja en proyectos más grandes o complejos donde los requisitos pueden cambiar a lo largo del tiempo. Además, debido a que la prueba y la depuración se realizan después de que se ha completado la implementación, los errores pueden ser costosos y difíciles de corregir.

### Fases

#### Fase de Requisitos

La primera fase del modelo en cascada es la fase de requisitos. En esta fase, los desarrolladores trabajan estrechamente con los clientes y los usuarios finales para recopilar y documentar todos los requisitos del software. Los requisitos pueden incluir funcionalidades, características, restricciones y objetivos del software. Esta fase es crucial para el éxito del proyecto, ya que cualquier malentendido o falta de claridad en esta etapa puede llevar a problemas en las fases posteriores.

#### Fase de Diseño

Una vez que se han recopilado y documentado todos los requisitos, el equipo de desarrollo pasa a la fase de diseño. En esta fase, se toman las decisiones sobre cómo se implementarán los requisitos en el software. Esto puede incluir la elección de la arquitectura del software, la selección de las tecnologías y herramientas que se utilizarán, y la creación de diagramas y modelos que ayuden a visualizar la estructura y el comportamiento del software.

#### Fase de Implementación

La fase de implementación es donde el diseño del software se convierte en código. Los desarrolladores utilizan el diseño y las especificaciones creadas en la fase de diseño para escribir el código del software. Esta fase también puede incluir la integración de diferentes componentes del software y la realización de pruebas unitarias para asegurarse de que cada parte del software funciona como se esperaba.

#### Fase de Verificación

Después de que el software ha sido implementado, entra en la fase de verificación. En esta fase, el software se somete a una serie de pruebas para asegurarse de que cumple con los requisitos establecidos en la fase de requisitos. Esto puede incluir pruebas de funcionalidad, pruebas de rendimiento, pruebas de seguridad y pruebas de usabilidad.

#### Fase de Mantenimiento

La última fase del modelo en cascada es la fase de mantenimiento. Una vez que el software ha sido lanzado y está en uso, es probable que surjan problemas o que se necesiten cambios. En la fase de mantenimiento, el equipo de desarrollo se encarga de corregir los errores, realizar mejoras y añadir nuevas características según sea necesario.

### Ventajas

#### Fácil de entender y usar

El modelo en cascada es simple y fácil de entender, lo que lo hace accesible para los equipos de desarrollo de software, especialmente para los que son nuevos en el desarrollo de software.

#### Estructura clara

El modelo en cascada proporciona una estructura clara y definida para el proceso de desarrollo de software. Cada fase tiene un conjunto específico de tareas y objetivos, lo que puede ayudar a mantener al equipo de desarrollo enfocado y organizado.

#### Documentación detallada

Debido a la naturaleza secuencial del modelo en cascada, se genera una gran cantidad de documentación en cada fase. Esto puede ser útil para la referencia futura y para la comunicación con los clientes y los usuarios finales.

### Desventajas

#### Difícil de cambiar

Una de las mayores desventajas del modelo en cascada es que es difícil hacer cambios una vez que el desarrollo ha avanzado más allá de la fase de requisitos. Esto puede ser problemático si los requisitos cambian o si se descubren problemas en las fases posteriores.

#### Retrasos en la entrega

Debido a que cada fase debe completarse antes de que pueda comenzar la siguiente, puede haber retrasos significativos en la entrega del software si hay problemas o retrasos en una fase.

#### No es adecuado para proyectos complejos o de larga duración

El modelo en cascada puede no ser la mejor opción para proyectos que son especialmente complejos o que se espera que duren mucho tiempo. En estos casos, un modelo de desarrollo más flexible, como el desarrollo ágil, puede ser más apropiado.