

Modelo de ciclo de Vida en Cascada

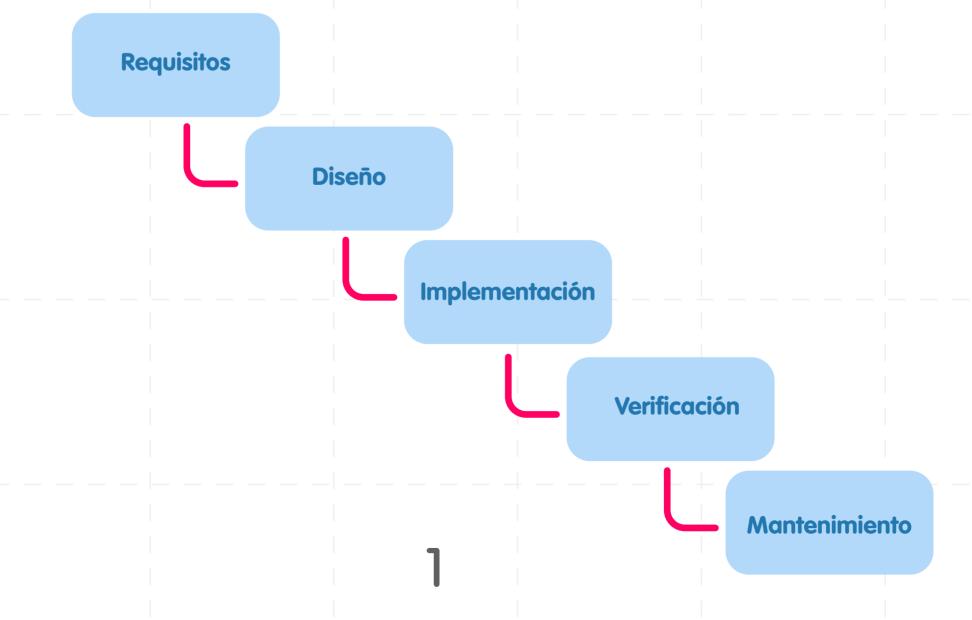


Hola:

El modelo en Cascada es el primer ciclo de vida y a pesar de que es fácil de entender y seguir, tiene muchos riesgos, lo que causó que prontamente fuera superado por otros ciclos de vida, pero este sigue siendo la base de muchos otros, así que usualmente se tiene como referencia.

El modelo de ciclo de vida en Cascada fue uno de los primeros que nació en el mundo del desarrollo de *software*, que consiste en un conjunto de fases secuenciales, es decir, cada fase solo se lleva a cabo cuando la anterior ha terminado.

Las fases que contemplan inician por lo general con requisitos para pasar a una fase de diseño y posteriormente sigue la fase de implementación o codificación, luego la verificación o pruebas y, por último, el mantenimiento. Sin embargo, la forma de representar esto permite ver que, por ejemplo, la fase de diseño solo se ejecuta después de que ha terminado la de requisitos. Una vez en esta fase no es posible regresar, por eso se llama Cascada.



Con un poco más de detalle se podría estar capturando requisitos, es decir, se interactúa con los clientes, de los que se debe especificar o documentar muy bien todo lo que dicen. Cuando ya se tiene todo muy claro, se pasa al diseño. Aquí por lo general se hacen diferentes modelos que se basan, por supuesto, en los requisitos que ha dado el usuario. Aquí se trata de definir cómo se va a implementar y cuando el diseño está terminado y se tengan claros todos sus detalles se puede pasar a la implementación, es decir, llevar al lenguaje de programación esa idea del diseño. Luego se hacen las pruebas o verificación que al concluir, se le entrega al usuario. Esto conformaría el ciclo de vida en Cascada.

Sin embargo, se deben notar ciertas cosas. El usuario solo interactúa al comienzo, en la parte de requisitos, y al final, cuando ya está haciendo las pruebas del *software*, es decir, que pasa mucho tiempo hasta que el usuario pueda ver algo terminado. Esto también es muy riesgoso, porque si en cualquiera de las fases se encuentra un problema, devolverse es complicado y costoso.

A diferencia de productos físicos como carros, casas, ropa u otros, el *software*, como no es tangible, es difícil que se defina por completo desde el comienzo. Lo más seguro es que al verlo funcionando aparezcan nuevos requisitos y se procure ajustar algunos de estos. Por lo tanto, este modelo en Cascada, aunque fue interesante en su comienzo, resulta bastante complejo ya que los gastos al usarlo suelen ser altos.

Ventajas +++

Es fácil de entender. Es un modelo que me permite entender las fases más importantes del desarrollo y su relación. Es fácil de planear y de controlar, porque se puede definir unos tiempos fijos para esas actividades.

Desventajas ---

No permite mostrar avances preliminares. El cliente espera mucho antes de ver el resultado final y no permite que se tenga realimentación temprana. Es muy rígido, no es fácil adaptarse a nuevas situaciones o inconvenientes. Si se presenta alguna dificultad o cambia el mercado (lo que es muy frecuente hoy en día) este ciclo de vida no sería apto para eso.

Es muy costoso y riesgoso de usar debido a la casi inexistente opción para modificar el proceso de este modelo. Este modelo puede ser empleado en software pequeño, poco complejo que tenga requisitos muy bien definidos

Ejercicio 🗸

Queda disponible un ejercicio: Investigar sobre el modelo de ciclo de vida llamado "Modelo en V". Es parecido al cascada, pero con algunas fases adicionales que tienen que ver con pruebas y calidad. ¿Cuáles son esas fases que tiene?, ¿cuáles son sus ventajas y desventajas? y ¿en cuáles casos se puede usar este modelo?

AAISION TIC2022



