DESCRIP

FILOSOFIA ADOPT Y PRINCIPIOS

PRESENTACION GRAFICA

El modelo en cascada: desarrollo secuencial de software

El término “waterfall model” hace referencia a un procedimiento secuencial que permite representar un proyecto en base a unas fases que se suceden entre sí.

Índice

[¿Qué es el modelo en cascada?](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#c178919)

[¿Cómo funciona el modelo en cascada?](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#c178921)

[Procedimientos alternativos al modelo en cascada](https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/#c178938)

Final del formulario

¿Qué es el modelo en cascada?

El desarrollo en cascada (en inglés, waterfall model) es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. El waterfall model se utiliza, especialmente, en el desarrollo de software.

¿Cómo funciona el modelo en cascada?

El desarrollo del modelo se atribuye al teórico de la informática Winston W. Royce. Sin embargo, Royce no es el inventor de este modelo. Muy al contrario, en su ensayo de 1970 titulado [Managing the Development of Large Software Systems](https://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf" \o "Ensayo disponible en la página web de la Universidad del Sur de California" \t "_blank), el teórico presenta una reflexión crítica acerca de los procedimientos lineales. A modo de alternativa, Royce presenta un modelo iterativo incremental en el que cada una de las fases se basa en la anterior y verifica los resultados de esta.

Royce propone un modelo compuesto por siete fases que se ha de ejecutar en diversas vueltas (iteraciones):

Requisitos de sistema

Requisitos de software

Análisis

Diseño

Implementación

Prueba

Servicio

El procedimiento popularmente conocido como waterfall model se basa en las fases definidas por Royce, pero solo prevé una iteración.

En el ensayo publicado por Royce, el término no aparece en ningún momento.

 Hecho

El modelo en cascada se popularizó a través de la norma estadounidense DoD-STD-2167. Esta norma se basa en una versión extremadamente simplificada del procedimiento desarrollado por Royce, que no fue lo suficientemente analizado por los autores. Tal y como reconoció con el paso del tiempo David Maibor, uno de sus autores, el motivo fue la falta de compresión de los modelos iterativos incrementales.

En la práctica, se aplican diversas versiones del modelo. Los más habituales son los modelos que dividen los procesos de desarrollo en cinco fases. En ocasiones, las fases 1, 2 y 3 definidas por Royce se integran en una sola fase de proyecto a modo de análisis de los requisitos.

Análisis: planificación, análisis y especificación de los requisitos.

Diseño: diseño y especificación del sistema.

Implementación: programación y pruebas unitarias.

Verificación: integración de sistemas, pruebas de sistema y de integración.

Mantenimiento: entrega, mantenimiento y mejora.

La siguiente imagen explica por qué el procedimiento lineal se denomina metodología en cascada.

[](https://www.ionos.es/digitalguide/fileadmin/DigitalGuide/Screenshots_2019/wasserfallmodell-ES-1.jpg)El modelo en cascada de cinco niveles, basado en las propuestas de Winston W. Royce, divide los procesos de desarrollo en las siguientes fases de proyecto: análisis, diseño, implementación, verificación y mantenimiento. El gráfico incluye una de las ampliaciones del modelo planteadas por Royce: la verificación de los resultados de cada una de las fases tomando en consideración las exigencias y especificaciones formuladas en el paso anterior.

En las ampliaciones de la metodología en cascada se añaden funciones iterativas al modelo básico como, por ejemplo, los saltos hacia atrás, que permiten comparar los resultados de cada una de las fases con las hipótesis obtenidas en la fase anterior, de modo que se puedan verificar.

Las fases del desarrollo en cascada

En este modelo, las diferentes fases de un proceso de desarrollo se suceden una detrás de otra como en una cascada. Cada una de las fases concluye con un resultado provisional (hito) como, por ejemplo, un catálogo de requisitos en forma de pliego de condiciones, la especificación de una arquitectura de software o una aplicación a nivel alfa o beta.

Análisis

Todo proyecto de software comienza con una fase de análisis que incluye un estudio de viabilidad y una definición de los requisitos. En el estudio de viabilidad se evalúan los costes, la rentabilidad y la factibilidad del proyecto de software. El estudio de viabilidad da como resultado un pliego de condiciones (una descripción general de los requisitos), un plan y una estimación financiera del proyecto, así como una propuesta para el cliente, si fuera necesario.

A continuación, se realiza una definición detallada de los requisitos, incluyendo un análisis de la situación de salida y un concepto. Mientras que los análisis de salida se encargan de describir la problemática en sí, el concepto ha de definir qué funciones y características debe ofrecer el producto de software para cumplir con las correspondientes exigencias. La definición de los requisitos da como resultado un pliego de condiciones, una descripción detallada de cómo se han de cumplir los requisitos del proyecto, así como un plan para la prueba de aceptación, entre otros.

Por último, la primera fase del waterfall model incluye un análisis de la definición de los requisitos en el que los problemas complejos se dividen en pequeñas tareas secundarias y se elaboran las correspondientes estrategias de resolución.

Diseño

La fase de diseño sirve para formular una solución específica en base a las exigencias, tareas y estrategias definidas en la fase anterior. En esta fase, los desarrolladores de software se encargan de diseñar la arquitectura de software, así como un plan de diseño detallado del mismo, centrándose en componentes concretos, como interfaces, entornos de trabajo o bibliotecas. La fase de diseño da como resultado un borrador preliminar con el plan de diseño del software, así como planes de prueba para los diferentes componentes.

Implementación

La arquitectura de software concebida en la fase de diseño se ejecuta en la fase de implementación, en la que se incluye la programación del software, la búsqueda de errores y las pruebas unitarias. En la fase de implementación, el proyecto de software se traduce al correspondiente lenguaje de programación. Los diversos componentes se desarrollan por separado, se comprueban a través de las pruebas unitarias y se integran poco a poco en el producto final. La fase de implementación da como resultado un producto de software que se comprueba por primera vez como producto final en la siguiente fase (prueba alfa).

Prueba

La fase de prueba incluye la integración del software en el entorno seleccionado. Por norma general, los productos de software se envían en primer lugar a los usuarios finales seleccionados en versión beta (pruebas beta). Las pruebas de aceptación desarrolladas en la fase de análisis permiten determinar si el software cumple con las exigencias definidas con anterioridad. Aquellos productos de software que superan con éxito las pruebas beta están listos para su lanzamiento.

Servicio

Una vez que la fase de prueba ha concluido con éxito, se autoriza la aplicación productiva del software. La última fase del modelo en cascada incluye la entrega, el mantenimiento y la mejora del software.

Pros y contras del modelo en cascada

Esta metodología permite estructurar la organización de forma clara en aquellos proyectos de desarrollo en los que las diversas fases de proyecto se diferencian claramente entre sí. Como cada una de las fases concluye con un hito, el proceso de desarrollo es muy fácil de comprender. El punto clave del modelo reside en la documentación de todos y cada uno de los pasos de proceso. Los conocimientos adquiridos se registran en pliegos de requisitos o borradores preliminares.

En teoría, el desarrollo en cascada pretende crear los requisitos previos para una ejecución rápida y rentable de los proyectos a través de una cuidada planificación previa. Sin embargo, la utilización del modelo en la práctica es controvertida. Por una parte, en el desarrollo de software las fases de proyecto no suelen estar claramente diferenciadas entre sí. Es precisamente en los proyectos de software más complejos donde los desarrolladores se suelen enfrentar al hecho de que los diversos componentes de una misma aplicación se encuentran en diferentes fases de desarrollo al mismo tiempo. Por otra parte, la secuencia lineal del waterfall model no suele coincidir con la realidad.

En sentido estricto, el modelo en cascada no prevé la realización de ajustes a lo largo del proyecto. Sin embargo, un proyecto de software en el que todos los detalles del desarrollo se definieran al comienzo, solo podría concluir con éxito si desde el principio se invirtiera una gran cantidad de tiempo y dinero en análisis y diseño. A todo esto, se añade que los proyectos de software de más envergadura se suelen prolongar durante varios años y, de no adaptarse a los avances más actuales, obtendrían resultados que ya estarían obsoletos en el momento de su aplicación.

Resumen de las ventajas y desventajas del modelo en cascada

| Ventajas | Inconvenientes |
| --- | --- |
| ✔ Una estructura sencilla gracias a unas fases de proyecto claramente diferenciadas. | ✘ Por norma general, los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas. |
| ✔ Buena documentación del proceso de desarrollo a través de unos hitos bien definidos. | ✘ Poco margen para realizar ajustes a lo largo del proyecto debido a un cambio en las exigencias. |
| ✔ Los costes y la carga de trabajo se pueden estimar al comenzar el proyecto. | ✘El usuario final no se integra en el proceso de producción hasta que no termina la programación. |
| ✔ Aquellos proyectos que se estructuran en base al modelo en cascada se pueden representar cronológicamente de forma sencilla. | ✘En ocasiones, los fallos solo se detectan una vez finalizado el proceso de desarrollo. |

La metodología en cascada se suele emplear, especialmente, en aquellos proyectos cuyos requisitos y procesos se pueden describir de forma precisa durante la fase de planificación, en los que cabe suponer que las hipótesis no van a sufrir una gran variación durante el transcurso del proyecto. Los procedimientos estrictamente lineales se adaptan, así, especialmente bien a proyectos de software pequeños, sencillos y claramente estructurados. A esta misma conclusión llegó Royce en los años 1970. Por este motivo, la alternativa al procedimiento lineal que propuso, y que más tarde se conocería como waterfall model, incluía tres ampliaciones principales:

Verificación tras cada fase de proyecto

Según Royce, los resultados de cada una de las fases de proyecto se deben comparar y verificar inmediatamente con los documentos elaborados previamente. Es decir, inmediatamente después de desarrollar un módulo, por ejemplo, se debería garantizar que este cumple con las exigencias definidas con anterioridad sin esperar a que concluya el proceso de desarrollo.

Al menos, dos iteraciones

Según Royce, el modelo se debería ejecutar en, al menos, dos ocasiones: primero para elaborar un prototipo y, a continuación, para desarrollar el producto de software en sí.

Pruebas que incluyen al usuario final

La tercera ampliación del modelo en cascada propuesta por Royce en su ensayo es una medida que, a día de hoy, se ha convertido en un procedimiento estándar en el desarrollo de productos: la inclusión del usuario final en el proceso de producción. Royce propone incluir al usuario en tres momentos diferentes del proceso de desarrollo de software: durante la planificación del software en la fase de análisis, entre el diseño del software y su implementación y en la fase de prueba que precede al lanzamiento del software.

Procedimientos alternativos al modelo en cascada

Debido a la secuencia estrictamente lineal de las fases sucesivas de proyecto, el modelo en cascada se adaptaría, en el mejor de los casos, a proyectos de software de poca envergadura. Por el contrario, los procesos complejos de varios niveles serían difícilmente representables con este modelo. Por este motivo, con el paso del tiempo han ido surgiendo enfoques alternativos.

Mientras que algunos modelos, como el modelo en espiral o el modelo en V, se consideran una evolución del waterfall model clásico, algunos métodos, como la programación extrema, el desarrollo ágil de software o el desarrollo iterativo, se centran en un enfoque completamente diferente y suelen permitir una adaptación a los cambios más recientes o a las exigencias más actuales.