

1) En las fases del ciclo de vida del software existen dos fases con nombre similar: implantación e implementación. ¿Podrías decir de qué fases estamos hablando, qué fases tienen antes y después y contextualizarlas con un ejemplo (imagina que diseñas una app para Iphone)?

La implementación hace referencia a la fase de codificación del producto, se realiza entre la fase de diseño y la fase de pruebas

La implantación consiste en la publicación de la solución final o la entrega del proyecto al cliente final y se realiza entre la fase de documentación y la fase de mantenimiento.

Por ejemplo, si se está desarrollando una app para iphone la implementación sería cuando se está codificando el producto mientras que la implantación es la fase en la que se trabaje para que dicha app se encuentre en la App Store.

2) En el ciclo de vida incremental existe una fase llamada integración. ¿Podrías buscar en la Red y en los apuntes y decirnos, con tus propias palabras, en qué consiste?

La fase de integración consiste en integrar todos los programas aprobados para implementarlos en la aplicación y así evaluarlo de forma conjunta.

FUENTE: [Las Fases del Modelo Incremental | Pasos para hacerlo \(comparasoftware.com\)](https://comparasoftware.com/2017/05/25/las-fases-del-modelo-incremental-pasos-para-hacerlo/)

3) El inicio de las metodologías ágiles se data el 12 de febrero de 2001, cuando 17 expertos firman lo que se llamó Manifiesto Ágil.

• FUENTE: <https://agilemanifesto.org/iso/es/manifesto.html>

Primero: Lista 12 principios que contiene (MANIFIESTO ORIGINAL)

Segundo: Intenta comprimirlos en 5 (MANIFIESTO RESUMIDO)

Sugerencia: Agrupa los puntos que hablan del equipo, de las entregas, del producto que se quiere obtener....

Por ejemplo: Si seleccionamos todos los puntos que hablan de entregas.

Tendremos unidos los puntos 1, 3 y 7 en un único punto

MANIFIESTO ORIGINAL:

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente. **(Entregas)**
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto. **(Equipo)**
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara. **(Entregas)**
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida. **(Equipo)**
9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad. **(Producto)**
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial. **(Producto)**
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

MANIFIESTO RESUMIDO:

1. Entregas funcionales, lo más frecuentes posibles, tempranas, continuas y con valor. (1,3,7)
2. Equipo motivado, auto-organizado y en continuo ajuste y perfeccionamiento. (5, 11, 12)
3. Adaptación a requisitos cambiantes y método eficiente y efectivo de comunicación. (2, 6)

4. Trabajo en equipo y desarrollo sostenible para mantener un ritmo constante. (4, 8)

5. Agilidad y simplicidad (9, 10)

• FUENTE: <https://agilemanifesto.org/iso/es/manifesto.html>

4) Enhorabuena, tu empresa ha firmado un contrato para realizar una aplicación de móvil por valor de 500.000 de euros, pero tu jefe sigue se empeña en usar el ciclo de vida en cascada. Ahora, en noviembre, cerrará los requisitos con el cliente y en julio tenéis prevista la entrega. Busca en la Red qué ocurrirá si el cliente cambia los requisitos iniciales en enero, es decir, si os escribe un email u os llama para deciros que hay ciertas funcionalidades que quiere modificarlas.

El proyecto debe iniciarse desde el principio con un nuevo código creando un sobrecoste o una pérdida de dos meses de trabajo en el desarrollo de la aplicación móvil.

5) Todas las fases del ciclo de vida del software son importantes, pero queremos que valores y analices la importancia de eliminar cada una de ellas del ciclo de vida de tu proyecto.

Existe una etapa que no puedes eliminar bajo ningún concepto ¿Cuál es esa etapa? ¿Crees que hay más de una “intocable”? No hay una única respuesta si la argumentas correctamente.

Análisis: Si eliminas esta etapa pueden entenderse mal las necesidades, los requisitos funcionales o no funcionales del proyecto que se desean cubrir, por tanto, usted asume el coste del desarrollo erróneo en el caso de que surgiera.

Diseño: Si eliminas esta etapa puede que no lleguemos a la mejor solución al problema que se nos propone, por lo tanto, asume el coste de encontrar más adelante la solución óptima.

Codificación o implementación: Si eliminas esta etapa no se codificará ni se realizará ningún elemento como son programas, bases de datos, servidores o comunicaciones, por lo tanto, va a tener que codificarlo por sus propios medios y el proyecto no podrá continuar hasta que se realice esta.

Pruebas: Si elimina esta etapa la aplicación no podrá probarse por una persona ajena a la fase de codificación y podrá contener errores, por tanto, usted asume el coste de solucionar estos errores (en caso de que surjan).

Documentación: Si elimina esta etapa la aplicación no será documentada por lo que, si el proyecto se tiene que paralizar y retomar, si hay que sustituir el equipo o se amplía el equipo no será tan ágil como si estuviera documentado, por tanto, usted asume ese sobrecoste en el caso de que lo hubiera.

Explotación o implantación: Si elimina esta etapa, la aplicación no será implantada por lo que no le llegará a ningún cliente interno o externo, ni será puesta en marcha, por tanto, estamos secuestrando su aplicación y mas adelante le pediremos un rescate por él.

Mantenimiento: Si eliminas esta etapa la aplicación podrá tener errores que no hayan sido localizados en la fase de prueba y no obtendrá actualizaciones por lo que podría dejar de ser funcional en futuras versiones del sistema operativo, por lo tanto, usted asume el coste de solucionar esos problemas (en caso de que surjan).

Después de dar un repaso a cada una de ellas considero que todas son necesarias en mayor o menor medida. Según el proyecto se podría invertir mas en unas que en otras pero todas ellas deben existir.