

DESARROLLO DE SOFTWARE

CODE UD3 – Tema 1

IES Plurilingüe Antón Losada Diéguez

Adrián Fernández González



Tabla de contenido

| | |
|--|---|
| 1. Introducción..... | 2 |
| 2. Desarrollo del software | 2 |
| 2.1. Planificación | 2 |
| 2.2. Implementación, pruebas y documentación | 3 |
| 2.3. Despliegue y mantenimiento | 3 |
| 3. Modelo en cascada o tradicional | 4 |
| 3.1. Ventajas | 4 |
| 3.2. Inconvenientes | 4 |
| 4. Modelo iterativo e incremental | 5 |
| 4.1. Ventajas | 5 |
| 5.2. Inconvenientes | 5 |
| 5. Modelo en espiral | 6 |
| 5.1. Determinar objetivos..... | 6 |
| 5.2. Análisis de riesgo..... | 6 |
| 5.3. Desarrollar y probar | 6 |
| 5.4. Planificación | 6 |
| 5.5. Ventajas | 6 |
| 5.6. Inconvenientes | 6 |
| 6. Kanban..... | 7 |
| 6.1. Ventajas | 7 |
| 6.2. Inconvenientes..... | 8 |
| 7. SCRUM..... | 8 |
| 7.1. Sprint | 8 |
| 7.2. Planificación | 8 |
| 7.3. Desarrollo..... | 8 |
| 7.4. Revisión..... | 9 |
| 7.5. Retroalimentación..... | 9 |
| 7.6. Ventajas | 9 |
| 7.7. Inconvenientes | 9 |

Desarrollo de software

1. Introducción

Con la aparición de los primeros lenguajes de alto nivel, se hizo más fácil programar, pero se empezó a descuidar mucho la eficiencia y seguía siendo difícil su seguimiento.

Una década más tarde, la situación es insostenible y varios expertos se reúnen para crear lo que posteriormente se denominaría ingeniería del software y la programación estructurada.

Detrás de la ingeniería del software está la idea de medir de forma sistemática y disciplinada el software, su desarrollo, funcionamiento y mantenimiento.



Desde su nacimiento, los ingenieros fueron mejorando los paradigmas, las herramientas para programar y los procesos necesarios para hacerlo. De esta forma, la programación se volvía, entre otras muchas cosas, más rápida, cómoda, legible, reutilizable y, en definitiva, óptima.

2. Desarrollo del software

Todo sistema ha de desarrollarse de una forma u otra, siguiendo un modelo u otro con sus etapas más o menos definidas y separadas.

Existen diversos modelos de programación, cada uno diseñado para un tipo de sistema, método de programación u objetivos. Cada equipo decide qué modelo usar y cómo, teniendo en cuenta las ventajas e inconvenientes de cada uno, así como de los recursos que disponen, el tiempo y las necesidades.

Cada modelo tiene sus peculiaridades, pero todos ellos tienen una serie de fases más o menos comunes.

2.1. Planificación

La primera parte es la obtención y análisis de requisitos. Aunque suele ser algo tedioso, es de vital importancia, puesto que a mayor descripción y más profunda, más sencillas se vuelven las siguientes etapas, pues da una idea clara de qué hacer y cómo.

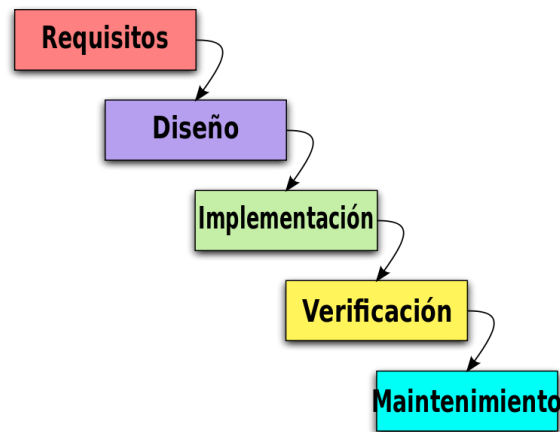
Hay que tener en cuenta que dentro de esta etapa también se encuentra una parte fundamental, enseñar a los usuarios finales cómo funciona y cómo utilizar el sistema. Habitualmente, estos se oponen al cambio, pues es más cómodo seguir con el estado actual que aprender uno nuevo.



Después de que todo esté correctamente funcionando, se realiza el mantenimiento, el proceso postventa en el que se van resolviendo posibles problemas, actualizaciones del sistema y aumento de funcionalidades a lo largo del tiempo.

3. Modelo en cascada o tradicional

Este modelo es el más básico y ampliamente utilizado para la realización de pequeños sistemas o por equipos que están empezando.



Tal como indica su nombre, es un proceso lineal en el que las etapas se suceden una detrás de la otra. La única etapa que se realiza en paralelo es la documentación.

3.1. Ventajas

- Coste reducido y poca carga de trabajo adicional.
- Fácil de entender e implementar.
- Conocido y usado con frecuencia.
- Establece una ruta eficiente, primero se planifica, luego se hace y por último se prueba.

3.2. Inconvenientes

- Rara vez algo se hace de forma lineal, siempre se suele ir a los poco realizando partes de forma independiente, diseñándolas, creándolas y probándolas en paralelo a otras partes.

- Si hubiese cualquier error en las primeras etapas, no se detectaría hasta las últimas, perdiendo gran cantidad de tiempo.
- Poca flexibilidad, todo es lineal y hay que esperar a terminar la fase anterior para pasar a la siguiente.

4. Modelo iterativo e incremental

Este modelo se basa en la idea de que todo se realiza en pequeños pasos cíclicos que van aumentando el sistema final, realizando las etapas una y otra vez hasta tener el sistema final.

El sistema se planifica en bloques o iteraciones en las que se realizan las distintas etapas y, al terminar ese bloque, se pasa al siguiente, repitiendo una y otra vez todos los procesos. Cada iteración se podría entender como un mini proyecto dentro del sistema.

Cada iteración debería tener un tiempo fijado corto, menor a un mes y ser todas de tiempos similares.

Este modelo se usa como idea base de otros muchos, denominados métodos ágiles, en contraposición a los más pesados y rígidos basados en el modelo en cascada.

4.1. Ventajas

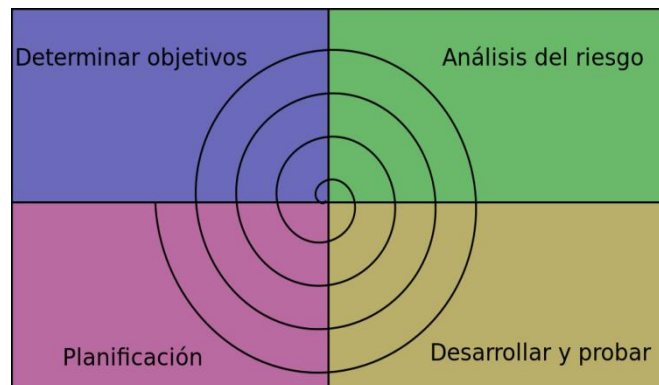
- Al realizar las pruebas continuamente, se detectan fallos antes y se pueden corregir más fácilmente.
- Permite mostrar al cliente resultados, mostrarle el progreso y dándole seguridad, obteniendo opiniones para realizar cambios.
- Los objetivos pueden ir adaptándose y especificándose poco a poco, reduciendo la carga de trabajo inicial y permitiendo adaptaciones y cambios.
- Si hubiese problemas de compatibilidad con el sistema final, se podrían detectar fácilmente en una iteración, pudiendo corregirse y no perder todo.
- Los pagos se pueden hacer secuenciales, ya que el cliente puede ver resultados y obtener sistemas más o menos funcionales.
- Permite centrarse en una parte en concreto cada vez, reduciendo la presión en el equipo.

5.2. Inconvenientes

- Generalmente, requiere una alta disponibilidad del cliente.
- Puede crear una falsa seguridad al cliente, que puede pensar que el sistema está terminado, dejándolo a medias.
- Se necesita más tiempo, puesto que hay que reunirse constantemente para establecer las iteraciones.

5. Modelo en espiral

Este modelo mezcla la idea del modelo en cascada con el iterativo e incremental, realizando iteraciones de cuatro fases pero cada vez más complejas y extensas.



Este modelo se divide en cuatro bloques fundamentales que han de realizarse en cada iteración.

5.1. Determinar objetivos

Se interactúa con el cliente para establecer los objetivos, alternativas, posibles soluciones y tecnologías a utilizar.

También se especifica la documentación a entregar.

5.2. Análisis de riesgo

Se analizan los posibles riesgos, fallos y problemas que puedan surgir, así como las alternativas y posibles soluciones.

5.3. Desarrollar y probar

Se decide el modelo a seguir para desarrollar, se implementa y se prueba. Esta etapa se puede realizar mediante el uso de otro modelo dependiendo de los riesgos y necesidades.

Se identifican posibles riesgos y problemas y se resuelven los anteriores.

5.4. Planificación

Se planifica la siguiente etapa en base a los fallos detectados, mejoras posibles y necesidades.

5.5. Ventajas

- Análisis de riesgos continuo, reduciendo su coste e impacto.
- Tiene en cuenta las mejoras del sistema durante la creación.
- Permite tener en cuenta la calidad del sistema.
- Une lo bueno de los otros modelos.
- Buena gestión de grandes proyectos.

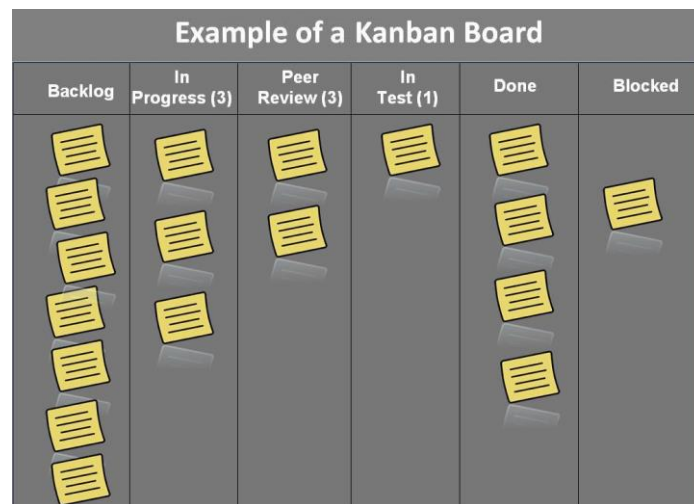
5.6. Inconvenientes

- El equipo ha de ser experto en el análisis y detección de riesgos.
- Genera un exceso de tiempo de análisis.
- Es más extenso y costoso que otros modelos.

- Es poco flexible, necesita seguirse a rajatabla o puede acarrear fallos.
- Diseñado para grandes proyectos, con el resto puede acarrear más inconvenientes que ventajas.

6. Kanban

Este modelo, del japonés Kan visual y Ban tarjeta, se basa en la idea de un tablón con tarjetas o post-it con las tareas a realizar y una descripción de cada una. El tablón se divide en las distintas etapas por las que pasa una tarea y en cada tarjeta estará localizada en el lugar en el que se encuentre esa tarea.



Cada vez que aparece una nueva tarea, ya sea un problema encontrado o un nuevo requisito, se crea una tarjeta y se posiciona al inicio, dando siempre prioridad a las tarjetas en curso.

A su vez, si el equipo es grande y hay varios departamentos, puede haber tableros para cada departamento, con sus propias fases y tarjetas, que irán desplazándose de tablón en tablón.

Este modelo ayuda a visualizar las partes y su estado de una forma clara, para organizarse más fácilmente, permitiendo priorizar tareas y asignarlas a las partes o equipos responsables.

Si esto se combina con tableros virtuales, webs dedicadas a ello o sistemas informáticos, su potencia se dispara, permitiendo asignación de tareas a los diferentes equipos, descripciones detalladas, interacción, etc.

Es una metodología utilizada no solo para el desarrollo del software si no que puede aplicarse para producción o gestión.

6.1. Ventajas

- Visualización rápida de las tareas y su estado.
- Facilita el priorizar las tareas y la dedicación enfocada a una tarjeta.
- Pueden combinarse varios proyectos en el tablón, ya que el equipo se centra en resolver las tarjetas, no un sistema.
- Detección rápida de embudos. Si una sección está llena de tarjetas, es un claro indicativo de que esa fase no es eficiente.

- Como cada tarea se realiza de forma independiente por todas sus fases, se prueban y se detectan los problemas continuamente.
- Al centrarse en una tarjeta, nos aseguramos de que esté terminada antes de pasar a la siguiente, sabiendo que esa parte está correcta.
- Es muy flexible, permitiendo combinar varios tableros y adaptarla a nuestras necesidades.
- Puede usarse junto a otras metodologías.
- Cada persona o equipo se centra en una tarea de cada vez, reduciendo el estrés y aumentando la concentración.
- Fácil de implementar, no es intrusivo y causa poco impacto.

6.2. Inconvenientes

- No se sabe a ciencia cierta el estado general del sistema.
- No hay fases del proyecto, por lo que no podemos hacer entregas periódicas a no ser que hagamos un tablón por etapa.
- Puede ser difícil usarla en proyectos grandes, por lo que suele usarse en conjunción con otros modelos.

7. SCRUM

Este modelo es el más utilizado en la actualidad y se basa en el desarrollo marcado por sprints, intervalos de tiempo reducidos en los que se desarrollan miniproyectos dentro del proyecto principal.

Este modelo está pensado para proyectos grandes que hay que realizar en tiempos reducidos y con gran incertidumbre.

7.1. Sprint

El sprint es la unidad de tiempo en el que se llevará a cabo un trabajo determinado.

Este ha de ser constante, predeterminada y entre dos y cuatro semanas, aunque se suele ajustar levemente a lo largo del proyecto.

7.2. Planificación

Se estipula el objetivo del sprint actual, plazos, funcionalidad, etc.

En esta fase se suele hablar con el cliente para establecer la priorización de objetivos y resolver dudas.

Esta reunión tiende a durar cuatro horas.

7.3. Desarrollo

Se realizan las operaciones necesarias, programación, pruebas, etc.

Cada día se hace una pequeña reunión de equipo en la que se estipula el trabajo diario y se comunica el estado de cada cosa, problemas encontrados, relación y dependencia entre las

partes, etc. Esta ha de ser breve, entre 5 y 15 minutos y solo tratar los puntos necesarios, sin discusiones ni debates.

En esta etapa no ha de modificarse el objetivo del sprint.

7.4. Revisión

Al final del sprint, se hace una reunión para evaluar entre todos los resultados del mismo, detectar puntos a cambiar y mejoras.

Es importante que participe todo el equipo, ya que cada uno da el punto de vista de su parte.

Esta reunión no ha de pasar las cuatro horas.

7.5. Retroalimentación

En esta etapa se habla con el cliente y los usuarios para obtener una retroalimentación del estado actual del proyecto. Esto permitirá adaptarse en el siguiente sprint.

Esta reunión tiende a durar cuatro horas.

7.6. Ventajas

- Dado que hay reuniones diarias, la comunicación y planificación es continua.
- Permite detectar problemas y adaptarse a los mismos de forma extremadamente rápida.
- Puede usarse de tal forma que cada sprint genere un sistema usable por el cliente, priorizando estas partes críticas.
- Al ser tan metódico y fijo, aumenta la calidad del resultado.
- Los tiempos están muy marcados, por lo que es bastante predecible a nivel de plazos de entrega.

7.7. Inconvenientes

- Requiere mucho tiempo y recursos.
- Las reuniones constantes pueden hacer que se pierda excesivo tiempo, sobre todo si el equipo no tiene las cosas claras.
- El equipo tiene que tener experiencia y madurez en el sector.
- Necesita un coordinador experto que gestione todo y se dedique casi de forma exclusiva a ello.
- Su poca flexibilidad puede acarrear muchos problemas si sucede algún error inesperado.