



DESARROLLO DE SOFTWARE



Unidad Didáctica 1

Contenidos

Bloques:

- I. El software del ordenador
 - A. Clasificación del Software
 - B. Software de Programación
- II. **Fases de desarrollo de una aplicación informática**
 - A. Ciclo de vida del software
 - B. **Modelos de ciclo de vida**

Ciclo de vida

El estándar ISO/IEC 12207-1 define **ciclo de vida del software como..**

- Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso

Ciclo de vida

- El ciclo de vida de un producto software comprende el periodo que transcurre desde que el producto es concebido hasta que deja de estar disponible o es retirado
- Se divide en **etapas o fases de desarrollo**:
 - ▣ Análisis
 - ▣ Diseño
 - ▣ Codificación
 - ▣ Pruebas
 - ▣ Documentación
 - ▣ Explotación
 - ▣ Mantenimiento

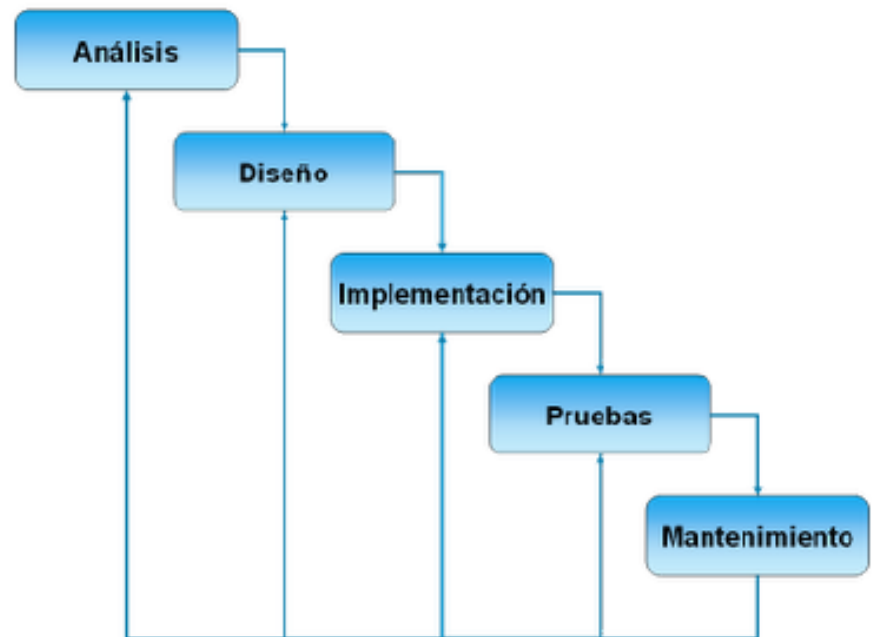
Modelos de ciclo de vida

- Existen varios modelos de ciclo de vida, es importante tener en cuenta las características del proyecto software para elegir un modelo u otro
- Los modelos más importantes son:
 - ▣ Cascada
 - ▣ Iterativo Incremental
 - ▣ Espiral
 - ▣ Métodos Ágiles

https://es.wikiversity.org/wiki/Procesos_de_desarrollo_software

Ciclo de vida en cascada

- En este modelo las etapas para el desarrollo del software tienen un orden, de tal forma que para empezar una etapa es necesario finalizar la etapa anterior
- Después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente



Ciclo de vida en cascada

- El modelo permite hacer iteraciones:
 - ▣ si el cliente en la etapa mantenimiento solicita un cambio en el diseño, se volvería a la etapa de diseño y se pasaría de nuevo por codificación y pruebas

Ventajas	Inconvenientes
<ul style="list-style-type: none">✓ Fácil de comprender, planificar y seguir✓ La calidad de producto resultante es alta✓ Permite trabajar con personal poco cualificado	<ul style="list-style-type: none">✓ Necesidad de tener todos los requisitos definidos al principio (normalmente el cliente no sabe que quiere al principio)✓ Difícil vuelta atrás si hay errores✓ El producto no está disponible para su uso hasta que no está terminado

Ciclo de vida en cascada

Se recomienda cuando:

- ❑ El proyecto es similar a alguno que se haya realizado con éxito anteriormente
- ❑ Los requisitos son estables y están bien comprendidos
- ❑ Los clientes no necesitan versiones intermedias

https://es.wikipedia.org/wiki/Desarrollo_en_cascada

Modelos Evolutivos de Ciclo de Vida

- Es usual que mientras se desarrolla el software los requisitos de los usuarios y del producto cambien
- Actualmente la gran competencia en el mercado y la urgencia de los usuarios para disponer del software obliga a desarrollar versiones del software no completas:
 - ▣ Versiones que posteriormente se irán perfeccionando y completando su funcionalidad
- En el modelo en cascada debemos entregar un producto completo, en cambio **el modelo evolutivo nos permite generar versiones cada vez más complejas** hasta llegar al producto final
- Los modelos evolutivos más conocidos son
 - ▣ **Iterativo incremental**
 - ▣ **Prototipos**
 - ▣ **Espiral**

Ciclo de Vida Iterativo Incremental

- Está basado en varios ciclos en cascada realimentados aplicados repetidamente
- Entrega el software en partes pequeñas pero utilizables, llamadas **incrementos**
- Cada incrementos se construye sobre aquel que ya ha sido entregado

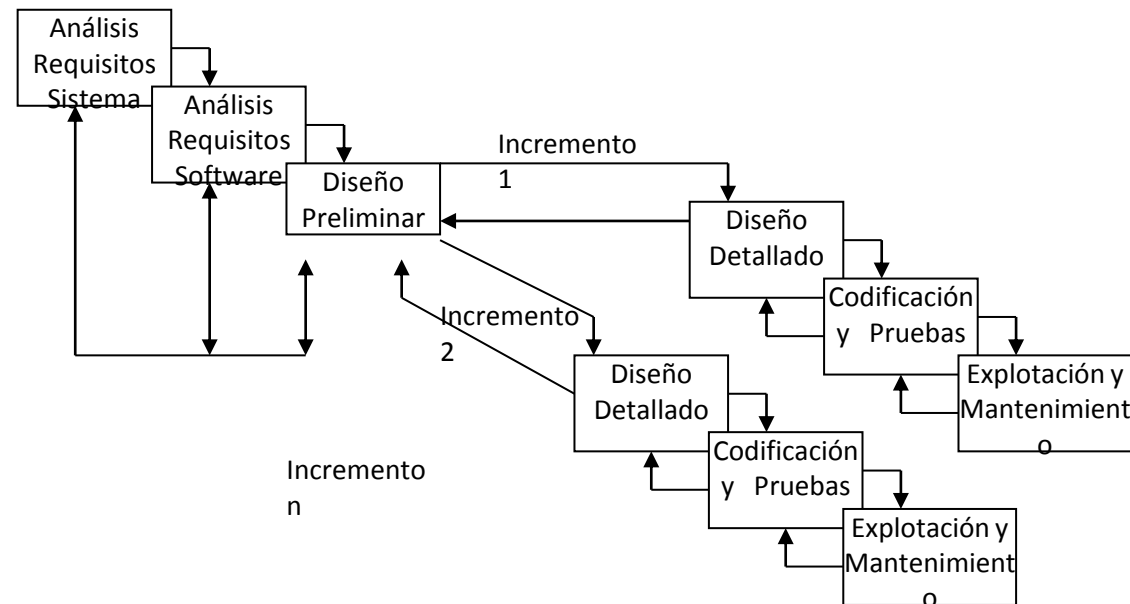


Figura 2. Modelo incremental

Ciclo de Vida Iterativo Incremental

Ventajas	Inconvenientes
<ul style="list-style-type: none">✓ No se necesitan conocer todos los requisitos al comienzo✓ Permite la entrega temprana al cliente de partes operativas del software✓ Las entregas facilitan la realimentación de los próximos entregables	<ul style="list-style-type: none">✓ Es difícil estimar el esfuerzo y el coste final necesario✓ Se tiene el riesgo de no acabar nunca✓ No recomendable para desarrollo de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido y/o de alto índice de riesgos

Se recomienda cuando

- Los requisitos o el diseño no están completamente definidos y es posible que haya grandes cambios
- Se están probando o introduciendo nuevas tecnologías

https://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente

Construcción de prototipos

Dos de las críticas que se hacían al modelo de ciclo de vida en cascada eran:

- es difícil tener claros todos los requisitos del sistema al inicio del proyecto
- no se dispone de una versión operativa del programa hasta las fases finales del desarrollo
 - ▣ esto dificulta la detección de errores
 - ▣ deja también para el final el descubrimiento de los requisitos inadvertidos en las fases de análisis.
- Para paliar estas deficiencias se ha propuesto un modelo de ciclo de vida basado en la construcción de prototipos
 - ▣ La construcción de prototipos se suele usar como parte de otros modelos

Construcción de prototipos

Prototipos: software con funcionalidad parcial, incompleta

- Se establece el desarrollo en **fragmentos** mas pequeños
 - ▣ Se reducen los riesgos inherentes del proyecto
 - ▣ Se logra que los cambios tengan menor impacto
- Pequeños prototipos con modificaciones son mostrados al cliente
 - ▣ Sirve para confirmar que se han comprendido sus requisitos
 - ▣ El usuario **involucrado** durante el desarrollo (probando prototipos) incrementa la aceptación de la implementación del producto final
- Muchos de los prototipos se generan con la expectativa de ser **descartados**, sin embargo, en algunos casos el prototipo puede evolucionar y convertirse en el producto final
- Es necesario un entendimiento fundamental de los problemas del negocio:
 - ▣ Evitar desarrollar prototipos que son prescindibles

Construcción de prototipos

Proyectos Candidatos:

- Cualquier aplicación que presente **mucha interacción con el usuario**, o que necesite algoritmos que puedan construirse de manera evolutiva, yendo de lo mas general a lo más específico es una buena candidata.
 - ▣ Pero si va a ser más costoso construir el prototipo que la aplicación final, no vale la pena
- **Para probar la eficiencia de los algoritmos** que se van a implementar, o **para comprobar el rendimiento de un determinado componente del sistema**
 - ▣ por ejemplo, una base de datos o el soporte hardware, en condiciones similares a las que existirán durante la utilización del sistema
- En otros casos, el prototipo servirá **para modelar y poder mostrar al cliente cómo va a realizarse la E/S de datos en la aplicación**,
 - ▣ El cliente puede hacerse una idea de como va a ser el sistema final, pudiendo entonces detectar deficiencias o errores en la especificación aunque el modelo no sea más que una cáscara vacía

Construcción de prototipos

Problemas:

- Con demasiada frecuencia el prototipo pasa a ser parte del sistema final
- El utilizar el prototipo en el producto final conduce a que éste contenga numerosos errores latentes, sea ineficiente, poco fiable, incompleto o difícil de mantener. En definitiva a que tenga poca calidad
- Otro problema sería que al sustituir el prototipo por el sistema final y desechar aquel perdiéramos toda la documentación existente sobre los requisitos del sistema si únicamente estaban recogidos en el prototipo y no existía otro tipo de documentación

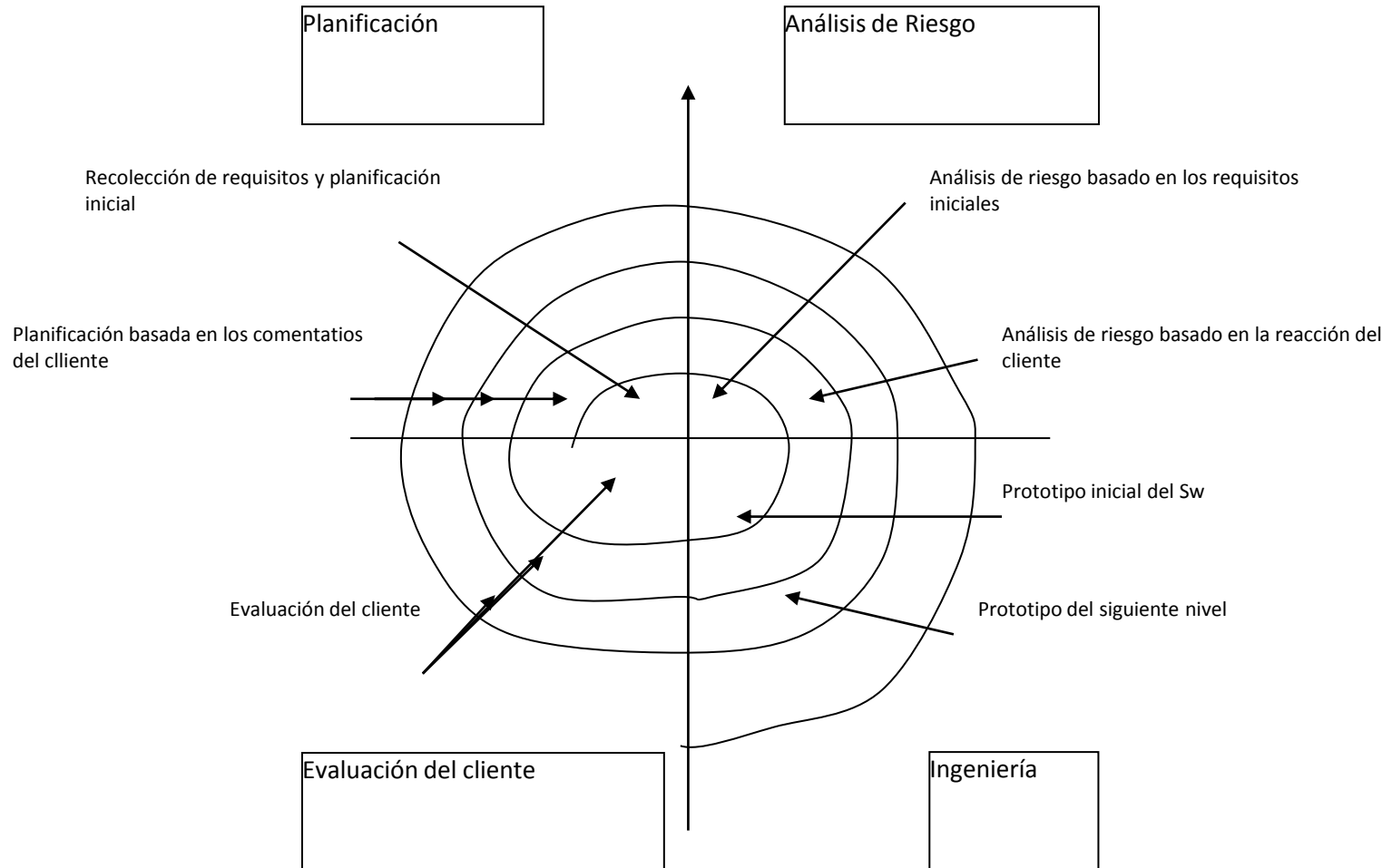
Ciclo de Vida en Espiral

- Este modelo combina el Modelo en Cascada con el modelo Iterativo con construcción de prototipos
- El proceso de desarrollo se representa como una espiral, donde en cada ciclo de la espiral se desarrolla una parte del mismo
- **Cada ciclo** está formado por cuatro fases y cuando termina **produce una versión incremental** del software con respecto al ciclo anterior
- En este aspecto se parece al Modelo Iterativo Incremental con la diferencia de que en cada ciclo se tiene en cuenta el **análisis de riesgos**

Ciclo de Vida en Espiral



Ciclo de Vida en Espiral



Ciclo de Vida en Espiral

Para cada ciclo, los desarrolladores siguen estas fases:

- **Determinar objetivos:** cada ciclo de la espiral comienza con la identificación de los objetivos, alternativas para alcanzar los objetivos (diseño A, diseño B, reutilización, compra, etc.) y las restricciones impuestas a la aplicación de las alternativas
- **Análisis de riesgos:** hay que evaluar las alternativas en relación con los objetivos y limitaciones. En este proceso se identifican los riesgos involucrados y (si es posible) la manera de solucionarlos
 - ▣ Un riesgo puede ser cualquier cosa: requisitos no comprendidos, mal diseño, errores en la implementación, etc.
- **Desarrollar y probar:** desarrollar la solución al problema en este ciclo y verificar que es aceptable
- **Evaluación-:** revisar y evaluar todo lo que se ha hecho, y con ello decidir si se continúa.
 - ▣ Si es así, hay que planificar las fases del ciclo siguiente

Ciclo de Vida en Espiral

Ventajas	Inconvenientes
<ul style="list-style-type: none">✓ No requiere una definición completa de los objetivos para empezar a funcionar✓ Análisis de riesgo en todas las etapas✓ Reduce riesgos del proyecto✓ Incorpora objetivos de calidad	<ul style="list-style-type: none">✓ Es difícil evaluar los riesgos✓ El costo del proyecto aumenta a medida que la espiral pasa por sucesivas iteraciones✓ El éxito del proyecto depende en gran medida de la fase de análisis de riesgos

Se recomienda cuando

- Proyectos de gran tamaño y que necesitan constantes cambios
- Proyectos donde sea importante el factor riesgo

https://es.wikipedia.org/wiki/Desarrollo_en_espiral

Ciclo de vida en espiral

La diferencia principal entre CVE y CP es

En el modelo de construcción de prototipos

- ▣ los prototipos se usan para perfilar y definir los requisitos
- ▣ Al final, el prototipo se desecha y comienza el desarrollo del software siguiendo el ciclo clásico

En el modelo en espiral

- ▣ los prototipos son sucesivas versiones del producto, cada vez más detalladas (el último es el producto en sí) y constituyen el esqueleto del producto de ingeniería
- ▣ Por tanto deben construirse siguiendo estándares de calidad

Desarrollo Ágil de Software

Tendremos una conferencia en el IES Abastos sobre Metodologías Ágiles y Extreme Programming.

Documéntate sobre el tema:

- https://es.wikiversity.org/wiki/Metodolog%C3%ADas_%C3%A1giles_de_desarrollo_software
- https://es.wikiversity.org/wiki/Extreme_Programming