

PRÁCTICA 1

INTRODUCCIÓN A LA PROGRAMACIÓN

Nombre: Hugo Pelayo

Asignatura: Programación

Curso: 2022-2023

Índice

Introducción	3
Modelos de desarrollo de software	4
Modelo en cascada	4
Modelo por prototipos	5
Modelo evolutivo.....	5
1. Modelo evolutivo exploratorio	6
2. Modelo evolutivo por prototipos desechables	6
Modelo incremental	7
Modelo en espiral.....	7
1. Objetivos y determinación de alternativas.....	8
2. Análisis y evaluación de riesgos	8
3. Desarrollo y fase de pruebas.....	8
4. Plan de siguiente iteración	9
Bibliografía.....	10

Introducción

Se define como ciclo de vida del software al conjunto de etapas por las cuales pasa un proyecto durante su "vida". Este proceso normalmente consta de etapas bien diferenciadas que son: **Especificación y Análisis de requisitos, Diseño del software, Codificación del software, Pruebas o Testes, Instalación y Producción, y Mantenimiento:**

1. **Fase de especificación:** En esta etapa se prepara el diseño y la implementación del producto o software. Aquí es necesario definir el margen del proyecto y optar por una metodología para su desarrollo. Es imprescindible también definir los plazos de tiempo para llevar a cabo ciertas tareas o reuniones que sean necesarias y designar los roles que le corresponderán a cada miembro del equipo de trabajo.
2. **Fase de análisis y requisitos:** Una escogida la metodología de desarrollo de software, se puede empezar por concretar una serie de requisitos que debería cumplir el software, estos determinan entre otros aspectos los cambios que son posibles de hacer en el proyecto si se optara por una actualización, por ejemplo.
3. **Fase de diseño del software:** En esta etapa se determina como el software debe cumplir con los requisitos aplicables. Conviene entonces, identificar variedad de soluciones posibles para el problema que se esté tratando, evaluarlas y determinar cuál de ellas es la más conveniente. Llegados a una solución se procede con optar por un hardware y software auxiliar como herramientas para desarrollar el proyecto.
4. **Fase de codificación del software:** Esta etapa marca básicamente el inicio de la implementación del software.
5. **Fase de pruebas y testes:** Una vez finalizada la etapa de codificación, nos queda realizar una serie de pruebas para determinar que el producto cumple con los requisitos establecidos. Se deben repetir las pruebas tantas veces como sea necesario para hasta que el cliente esté conforme con el producto final.
6. **Fase de instalación y producción:**

Esta es la parte más importante de todas las fases de desarrollo del software y la más duradera. Su mantenimiento incluye tres puntos importantes:

- Eliminar defectos durante su uso, que se conoce con el nombre de mantenimiento correctivo.
- Adaptarlo a nuevas necesidades, conocido como mantenimiento adaptativo.
- Añadirle nuevas funcionalidades, conocido también con el nombre de mantenimiento perfectivo.

Cuanto mejor sea el software, más tiempo se acostumbra a invertir en su mantenimiento ya que cuando se llega a hacer uso intensivo del software, cabe la posibilidad de surgir propuestas de mejora.

Modelos de desarrollo de software

Se conoce como metodología de desarrollo de software en el ámbito de la ingeniería del software al proceso de estructurar, planificar y controlar el desarrollo de sistemas de información.

Los métodos de desarrollo de software tienen como objetivo presentar un conjunto de técnicas modernas que nos permitan desarrollar software de calidad. Para esto acostumbran a utilizar herramientas de análisis y diseño de objetos, como es el caso del lenguaje UML. Estos modelos se llevan a cabo a través del ciclo de vida del software mencionado anteriormente.

Modelo en cascada

Es una metodología secuencial. En ella las etapas de desarrollo del software se “miran hacia abajo”, es decir, en forma de cascada de agua mediante las fases de análisis de requisitos, el diseño del producto, implementación, pruebas (comprobación de la validez del producto), la integración y mantenimiento. La primera definición formal de esta metodología se recoge en un artículo de Winston Royce W. publicado en el año 1970, aun así, Roy no hace referencia a este método con el nombre de “cascada”.

Las etapas de esta metodología siguen un orden riguroso. Antes de cada una, el equipo de desarrollo revisa que el producto esté listo y cumpla con los requisitos necesarios para superar la fase actual y pasar a la siguiente. Los requisitos de cada fase no son propensos a cambios, esto hace del procedimiento cascada poco flexible ya que los requisitos acostumbran a recogerse cuando se plantea el proyecto, en ese momento se recoge información como el coste, riesgos de producción, intervalos de fechas límite para cada fase del proyecto, entre otros aspectos.

El modelo en cascada tiene como principios básicos los siguientes:

- Se divide en fases secuenciales, con cierta superposición entre ellas.
- Se enfatiza a la planificación, los horarios de desarrollo activo, los intervalos de fechas límite entre fases intermedias, presupuestos.
- Se controla el proyecto de manera estricta, a través de un escrito, que contiene información detallada del producto.

Este modelo lo adoptan administradores de proyectos para el desarrollo de proyectos que no tienen requisitos ambiguos (el proyecto está claramente definido), proyectos en los cuales hay un trayecto claro de cómo se procederá con el desarrollo o simplemente el cliente sabe el administrador que el cliente con muy poca probabilidad solicitará cambios sobre el producto cuando este se encuentre en desarrollo.

Entre las principales desventajas del modelo en cascada podemos destacar los siguientes aspectos:

- Los proyectos toman largo tiempo en realizarse.
- Con frecuencia, los clientes no están seguros del todo de como quieren el producto final, podrían solicitar cambios durante el desarrollo de este.
- El diseño e implementación del producto es ajeno a los clientes.
- Efectos secundarios en las fechas límite de cada una de las etapas, es decir, si una fase del proceso se retrasa, las fases posteriores sufren de lo mismo.

Entre las principales ventajas del modelo en cascada podemos distinguir los siguientes aspectos:

- Los desarrolladores pueden descubrir errores de diseño durante el análisis del proyecto.
- El coste total del proyecto no fluctúa, es fijo desde el principio.
- Los desarrolladores de nuevo ingreso al proyecto pueden entender fácilmente el proyecto ya que toda la información sobre este está recogida en un escrito.
- Los clientes no solicitan nuevas funcionalidades, por tanto, el tiempo de producción en un principio es fijo, y plazo de entrega del producto final puede estar garantizado.

Modelo por prototipos

Es una metodología de desarrollo de software que permite poner a prueba el proyecto en sin haberlo acabado. Para entenderlo es importante saber que es un prototipo, que no deja de ser nada más que una versión prematura de un sistema de información. En esta metodología se crea una figura inacabada del producto con la finalidad de aprender sobre los requerimientos del producto final. La ventaja que conlleva es que el proceso de crear un prototipo es fácil y se puede llevar a cabo en muy poco tiempo.

En este modelo, el prototipo es entregado al cliente con el fin de que este experimente con él. Los desarrolladores entonces pueden recibir *feedback* de los usuarios del producto y determinar qué aspectos de este necesitan mejora.

La metodología por prototipos ha sido utilizada con bastante frecuencia desde los años 90, por que facilita poder definir los requisitos de software más complejos, esto es muy importante ya que para muchos clientes consideran importante proveer retroalimentación basada en la utilización del software, esto ayuda con definir de una manera más precisa los detalles del producto.

A diferencia del modelo evolutivo, los prototipos se construyen a partir de los aspectos más translúcidos del proyecto, esto con el hecho de poder expresar al máximo las críticas del cliente y saber que funcionalidades añadirle al producto y de cuales deshacerse. Cabe destacar que unos prototipos pueden llegar a ser completamente descartados y otro a formar parte del producto final, dependiendo de cómo se ajusten a las necesidades del cliente.

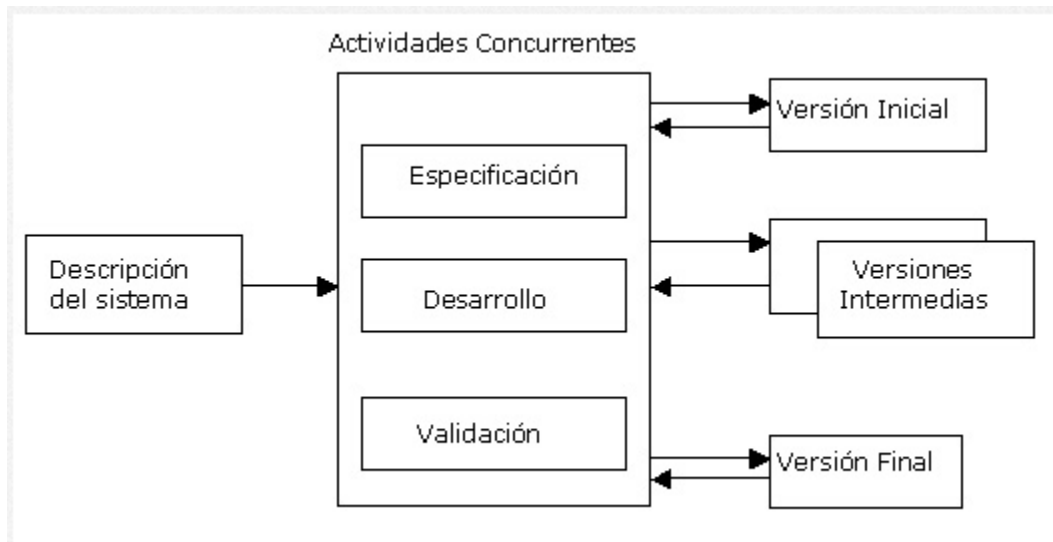
Esta metodología es útil cuando los requisitos varían con frecuencia (un producto muy flexible durante su desarrollo) o cuando se desconoce muy bien la aplicación. Importante destacar también que esta metodología de desarrollo es ágil y muy útil cuando se quiere probar alguna nueva arquitectura o tecnología.

Como desventajas de esta metodología, podemos destacar que no se conoce cuando tendría un software en condiciones de prueba, no se sabe cuántos prototipos serán necesarios para llegar al resultado final del proyecto.

Modelo evolutivo

El desarrollo mediante metodología por evolución se basa en una versión inicial del producto que una vez expuesto se va refinando de acuerdo con la opinión que se recibe del cliente, tenemos unidas entonces las fases de especificación, desarrollo y validación, en vez de estar separadas que sería lo común en otras metodologías. Consta de siete fases bien diferenciadas: comunicación, analogía, plan rápido, modelado y diseño rápido, construcción del prototipo, desarrollo junto con

entrega y *feedback* del cliente, y, para acabar, una entrega final. Este modelo fue propuesto en un principio por Barry Boehm.



Podemos destacar dos tipos de metodologías evolutivas:

1. Modelo evolutivo exploratorio

En este caso el objetivo es trabajar de manera estrecha con el cliente para explorar los requisitos del producto. El desarrollo empieza con las partes del software que estén claramente definidas, a partir de esa versión, se va añadiendo nuevas funcionalidades, así como corregir erros según convenga.

2. Modelo evolutivo por prototipos desechables

En este tipo de modelo evolutivo, el objetivo es comprender los requisitos del cliente y entonces desarrollar una definición mejorada de los requisitos del software. A diferencia del modelo evolutivo anterior en este caso, se comienza con un prototipo que resultado de los requisitos del cliente que no quedan bien definidos.

La metodología evolutiva acostumbra a aportar más ventajas que desventajas en comparación con la metodología por cascada ya que se adapta con más facilidad a las necesidades del cliente a la vez que este entiende mejor su producto y puede refinarlo con más facilidad. Sin embargo, el método evolutivo tienes dos problemas principales. Por una parte, los administradores tienen que hacer entregas regulares para notificar al cliente de los avances hasta el momento, si los prototipos se desarrollan muy rápidamente se vuelve poco rentable documentar la información de cada versión. Por otra parte, los cambios continuos sobre el producto dificultan cada vez más la evolución de éste.

Modelo incremental

En esta metodología se construye el producto final de manera progresiva, provee estrategias para controlar la complejidad del software y evitar riesgos de desarrollo, reservando parte del desarrollo para futuras fases.

En cada etapa del modelo incremental se agrega una nueva funcionalidad al software, este permite discernir cambios hechos en cada fase y ver los avances del producto de forma más rápida en comparación con el modelo en cascada. Esta metodología es mucho más flexible que las demás y por lo general se suele poder utilizar el software antes de haber obtenido el resultado final.

Este modelo de desarrollo de software surge como alternativa al método de desarrollo por cascada.

Principales ventajas:

- La solución mejora de forma progresiva a través de reiteradas reformas del producto, con la retroalimentación del cliente se entiende mejor el producto y se refina más el producto final
- Los clientes no deben esperar hasta la entrega final del producto para hacer uso del software.
- Los clientes mantienen estrecha comunicación con el equipo de desarrollo para aportar información sobre detalles del producto y mejorar el entendimiento de este.
- Disminuye el riesgo de fracasar durante el proyecto, ya que desde la primera entrega se tiene un software usable. Las entregas acostumbran a ser fáciles de desarrollar.

Principales desventajas:

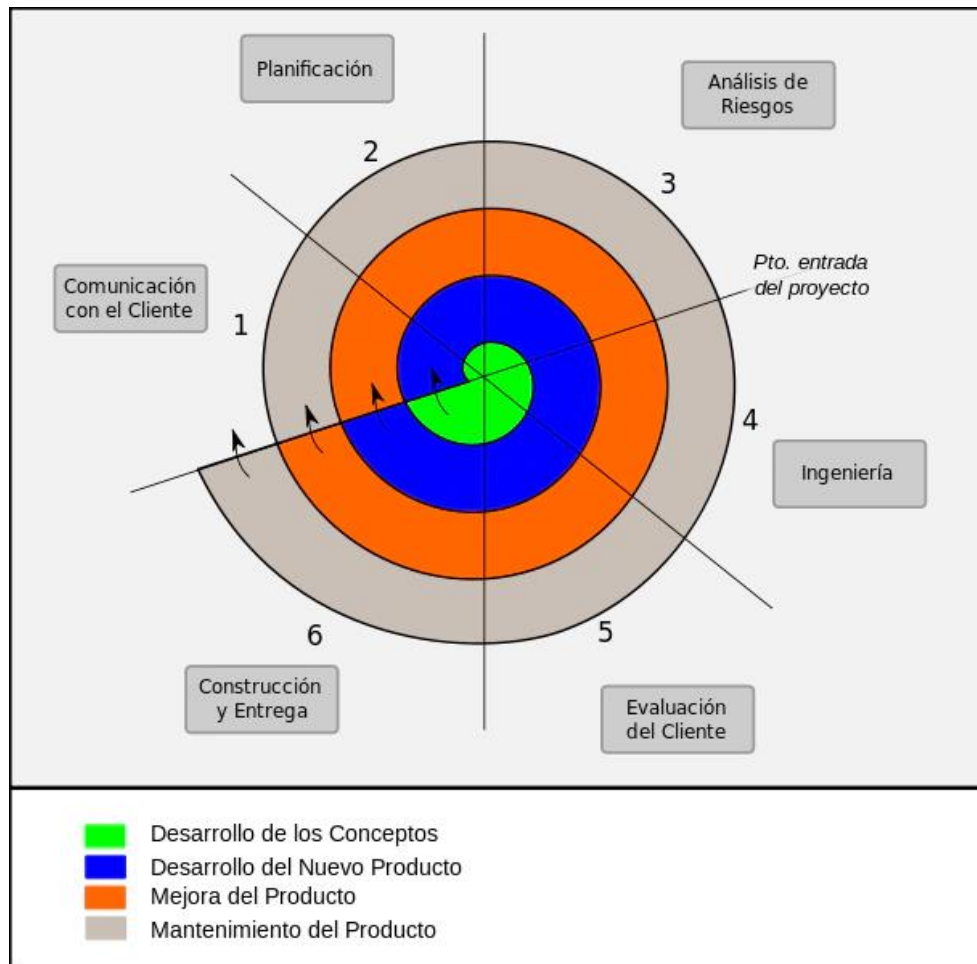
- Requiere de mucho planteamiento.
- Se requiere de objetivos claros para saber con claridad el estado del proyecto.

Modelo en espiral

La metodología de desarrollo de software en espiral es una metodología que resulta de la combinación del modelo de desarrollo por prototipos y el modelo de desarrollo por cascada, a su vez, añade el concepto de análisis de riesgos. El nombre viene dado por cómo se ejecutan sus fases, en forma de espiral. Cuanto más lejos del centro se encuentra en el desarrollo del producto, más avanzado está. En este modelo el producto se suele trabajar de manera continua y las mejoras vienen dadas a través de avances muy pequeños en el proceso de desarrollo.

Como se ha mencionado con anteriormente el modelo en espiral ofrece el análisis de riesgos, en efecto, enfatiza la minimización de los mismo, cosa que puede resultar en aumento del presupuesto de producción total, un esfuerzo mayor y posiblemente lanzamientos entre fases muy retardados. Sin embargo, estos aspectos se contrarrestan con enfoque incremental, empezando el proyecto por prototipos, estos prototipos son los que se acostumbran a refinar a través de las diferentes fases de desarrollo.

Este modelo fue propuesto por Barry Boehm W. en su ensayo "A Spiral Model of Software Development and Enhancement", en una época en la cual la metodología por cascada prevalecía. En este modelo destaca el análisis y la prevención de riesgos de desarrollo.



El modelo en espiral consta de varias fases de desarrollo propias:

1. Objetivos y determinación de alternativas

Los objetivos se determinan en conjunto con el cliente, a la vez que se discuten posibles alternativas y especifican condiciones marco, por ejemplo, los sistemas operativos para el desarrollo, entornos de desarrollo, lenguajes de programación, entre otras.

2. Análisis y evaluación de riesgos

Se descubren y evalúan los posibles riesgos de desarrollo. A parte de eso, también se consideran las alternativas existentes. Los riesgos se registran y reducen mediante la utilización de prototipos, simulaciones y software de análisis.

3. Desarrollo y fase de pruebas

Se añaden funcionalidades nuevas a los prototipos para ampliarlos.

4. Plan de siguiente iteración

El siguiente ciclo se planifica al final de cada fase. Si hay errores de desarrollo se buscan soluciones, si surge una mejor alternativa, si implementa en la fase posterior del ciclo de desarrollo.

El modelo en espiral se utiliza para proyectos más grandes que están sujetos a bastantes riesgos, ya que estos riesgos tienen un impacto importante en el desarrollo del producto y el control de costes para los clientes. También se utiliza en entornos técnicos más modernos ya que estos están sujetos a riesgos.

Los conflictos de diseño se evitan a través del enfoque en espiral, ya que se pueden testear con frecuencia y hacer los cambios necesarios de manera eficaz.

Ofrece la posibilidad de obtener *feedback* de parte del cliente u otros usuarios del producto como desarrolladores de las fases prematuras del producto. Sin embargo, este proceso requiere de una gestión que tiene en cuenta las fases de desarrollo y puede responder rápidamente a los riesgos. Requiere entonces de documentación para registrar los cambios hechos hasta el momento.

Bibliografía

- [1] Wikipedia, «Wikipedia - Metodología de desarrollo de software,» 11 Septiembre 2022. [En línea]. Available: https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software. [Último acceso: 3 Octubre 2022].
- [2] R. L. Ruiz, UT1.5 - Ciclo de vida del Software, Torrejón de Ardoz: Ruth Luspitao Ruiz, 2022.
- [3] I. NEWS, «Ciclo de vida del software,» 28 Noviembre 2020. [En línea]. Available: <https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>. [Último acceso: 7 Octubre 2022].
- [4] A. C. Team, «Waterfall Methodology,» 18 Marzo 2022. [En línea]. Available: <https://business.adobe.com/blog/basics/waterfall>. [Último acceso: 8 Octubre 2022].
- [5] S. Universidades, «Metodologías de de desarrollo de software,» 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 10 Octubre 2022].
- [6] U. -. U. C. d. Argentina. [En línea]. Available: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>. [Último acceso: 8 Octubre 2022].
- [7] Weebly, «INGENIERIA DE SOFTWARE,» [En línea]. Available: <https://ingsoftware.weebly.com/modelo-evolutivo.html>. [Último acceso: 2022 Octubre 9].
- [8] Organosdepalencia, «¿Qué es el modelo evolutivo?,» 23 Diciembre 2021. [En línea]. Available: <https://organosdepalencia.com/biblioteca/articulo/read/174366-que-es-el-modelo-de-evolutivo>. [Último acceso: 9 Octubre 2022].
- [9] RyteWiki, «Modelo en Espiral,» [En línea]. Available: https://es.ryte.com/wiki/Modelo_en_Espiral. [Último acceso: 9 Octubre 2022].