

Modelos del desarrollo de software

En los años 50 no existían metodologías de desarrollo, el desarrollo estaba a cargo de los propios programadores. No se sabía la fecha exacta en que concluiría un proyecto de software, no había forma de controlar las actividades que se estaban desarrollando. Tampoco se contaba con documentación estandarizada. El nacimiento de técnicas estructuradas es lo que da origen al desarrollo de aplicaciones a través de métodos de ingeniería. La informática aporta herramientas y procedimientos que se apoyan en la ingeniería de software con el fin de mejorar la calidad de los productos, aumentar la productividad y trabajo de los ingenieros desarrolladores de software, facilitar el control del proceso de desarrollo de software y suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.

El objetivo principal que busca la ingeniería de software es convertir el desarrollo de software en un proceso formal, con resultados predecibles, que permitan obtener un producto final de alta calidad y satisfaga las necesidades y expectativas del cliente.



Modelos del desarrollo de software

Una metodología define una representación que permite facilitar la manipulación de modelos y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema.

Goncalves (2005) plantea que la experiencia ha demostrado que los proyectos exitosos son aquellos que son administrados siguiendo una serie de procesos que permiten organizar y luego controlar el proyecto, considerando válido destacar que aquellos procesos que no sigan estos lineamientos corren un alto riesgo de fracasar.

Metodología de desarrollo de software: es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos.

Los modelos pueden pensarse como marcos de trabajo del proceso y que pueden ser adaptados para crear procesos más específicos.



Modelo de cascada

El desarrollo en cascada 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.

Royce propone un modelo compuesto por siete fases:

1. Requisitos de sistema
2. Requisitos de software
3. Análisis
4. Diseño
5. Implementación
6. Prueba
7. Servicio



Modelo de cascada



Modelo de cascada

- 1) Análisis y definición de requerimientos:** Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios.
- 2) Diseño del sistema y del software:** Se establece una arquitectura completa del sistema, el diseño del software identifica y describe los elementos abstractos que son fundamentales para el software y sus relaciones.
- 3) Implementación y pruebas unitarias:** Durante esta etapa, el diseño del software se lleva a cabo como un conjunto de unidades de programas, la prueba unitaria implica verificar que ésta cumpla su función específica.
- 4) Integración y prueba del sistema:** Los programas o las unidades individuales de programas se integran y se prueban como un sistema completo, para así asegurar que se cumplan los requerimientos.
- 5) Funcionamiento y mantenimiento:** En esta fase el sistema se instala y se pone en funcionamiento práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores así como la implementación de nuevos requerimientos.



Modelo de Prototipos

El modelo de prototipos permite que todo el sistema o algunas de sus partes, se construyan rápidamente para comprender con facilidad y aclarar ciertos aspectos en los que se aseguren que el desarrollador, el usuario y el cliente estén de acuerdo en lo que se necesita, así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo.

Este modelo se utiliza para dar al usuario una vista preliminar de parte del software. Se basa en prueba y error, ya que si al usuario no le gusta una parte del prototipo, significa que la prueba falló por lo cual se debe corregir hasta que el usuario quede satisfecho.

El prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero.

El construir el prototipo nos asegura que nuestro software sea de mejor calidad, además de que su interfaz gráfica sea del agrado del usuario.



Modelo de Prototipos

Hay dos clases de prototipos:

- Desechable: Nos sirve para eliminar dudas sobre lo que realmente quiere El cliente, además para desarrollar la interfaz que más le convenga al Cliente.
- Evolucionario: Es un modelo parcialmente construido que puede pasar de Ser prototipo a ser software pero no tiene una buena documentación y calidad.



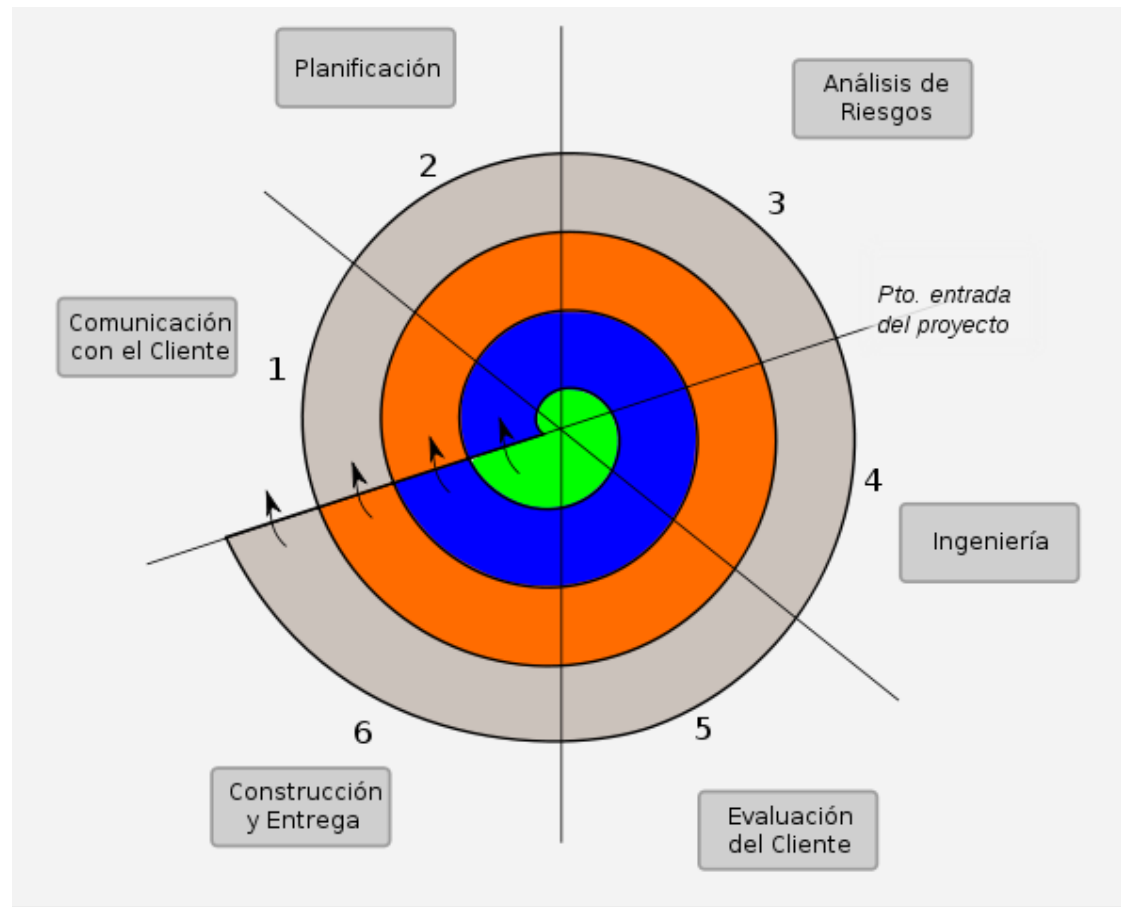
Modelo de Espiral

Este modelo describe el ciclo de vida de un software por medio de espirales que se repiten hasta que se puede entregar el producto terminado. El desarrollo en espiral también se conoce como desarrollo o modelo incremental. El producto se trabaja continuamente y las mejoras a menudo tienen lugar en pasos muy pequeños. Cuando se aplica este modelo, el software se desarrolla en una serie de entregas evolutivas. Cada una de las actividades del marco de trabajo representan un segmento de la ruta en Espiral.

Este modelo se basa en la idea de desarrollar una implementación inicial, exponiéndola a los comentarios del usuario y refinándola a través de las diferentes versiones que se generan hasta que se desarrolle un sistema adecuado.



Modelo de Espiral



- Desarrollo de los Conceptos
- Desarrollo del Nuevo Producto
- Mejora del Producto
- Mantenimiento del Producto



Modelo ágil

Es un marco de trabajo conceptual de la ingeniería de software que promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Existen muchos métodos de desarrollo ágil, la mayoría minimiza riesgos desarrollando software en cortos lapsos de tiempo. Se caracterizan por enfatizar la comunicación frente a la documentación, Por el desarrollo evolutivo y por su flexibilidad.

Elementos clave

- **Individuos:** Toda la importancia hay que dársela a las personas, que deben Permanecer en un primer plano.
- **Software funcionando:** Esto se debe a que había llegado un punto en el que la Documentación de un trabajo había alcanzado tanta importancia como el objeto De trabajo en sí mismo, el producto. Cuando realmente la mayor atención Debe estar puesta siempre en lo que queremos construir y lo demás debería Ser secundario.



Modelo ágil

- **Colaboración del cliente:** A la hora de sacar un proyecto adelante, la forma más productiva siempre será estableciendo un marco de colaboración y confianza con quién nos lo encarga. Tanto cliente como el desarrollador comparten objetivos e intereses similares.
- **Respuesta al cambio:** Se trata de apreciar la incertidumbre como un componente básico del trabajo, de tal manera que la adaptabilidad y la flexibilidad se convierten en virtudes y no en defectos de la manera de trabajar del equipo.

Éstos modelos deben cumplir con lo siguiente:

- Ser adaptables de forma incremental
- Tener abundante retroalimentación del cliente
- Basarse en la entrega continua de incrementos

