|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelo de Desarrollo de Software | Definición | | Características | Etapas |
| Modelo Prototipo | También conocido como desarrollo con prototipación o modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Este modelo se utiliza para dar al usuario una vista preliminar de parte del software.   Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. | | * No modifica el flujo del ciclo de vida * Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios * Reduce costo y aumenta la probabilidad de éxito * Exige disponer de las herramientas adecuadas * Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. * También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina. | * Recolección y refinamiento de requisitos * Modelado, diseño rápido * Construcción del Prototipo * Desarrollo, evaluación del prototipo por el cliente * Refinamiento del prototipo * Producto de Ingeniería |
| Cascada | El modelo en cascada o waterfall model, es la propuesta de un enfoque metodológico que consiste en ordenar de forma lineal las distintas etapas que debes de seguir al momento de desarrollar tu software. | | * Te ayuda a llevar un orden y organizar tu trabajo. * Es muy útil si no tienes demasiada experiencia. * Funciona de manera óptima en la mayoría de los dispositivos. * Es sencillo y fácil de seguir. * El modelo en cascada te ayuda a tener claridad en tus objetivos desde el comienzo del proyecto. * Al encontrar un problema, puedes fácilmente detectar la fase en la que surgió y así arreglarlo más rápidamente. * Si estás realizando un proyecto grande o muy complejo, puede que sea más difícil dividirlo en fases ordenadas, por lo que este sistema puede no ser el más adecuado. * Debido a la forma de trabajo lineal, tienes menos tiempo para concluir cada una de las etapas del modelo en cascada. * No puedes pasar a la etapa siguiente hasta que completes la anterior. * En ocasiones, los fallos no se detectan hasta la última fase del desarrollo, por lo que, para resolverlo tendrás que regresar a las fases anteriores y repetirlas o modificarlas. | * Análisis * Diseño * Implementación * Verificación * Mantenimiento |
| Dientes de Sierra | El modelo de diente de sierra muestra las percepciones del sistema por parte del usuario y el desarrollador de software en diferentes niveles de abstracción a lo largo del tiempo.  Al inicio del proyecto los desarrolladores y los usuarios están en el mismo nivel de abstracción, es decir, los requerimientos del sistema como se describen en el enunciado del problema | | Es un modelo V modificado que incluye intersecciones para evaluar cada uno de los prototipos de las etapas. Se le llama modelo de diente de sierra porque cada demostración del prototipo da como resultado un diente. La punta de cada diente es una intersección con el nivel de abstracción del cliente. |  |
| Rational Rose | Es una herramienta CASE (Computer – Arded Software Engineering), traducido al español como Ingeniería Asistida por Computadora, desarrollada por Rational Corporation basada en el Lenguaje Unificado de Modelación (UML), que permite crear los diagramas que se van generando durante el proceso de Ingeniería en el Desarrollo del Software. | | Entre las características principales de Rational se pueden destacar:   * Admite como notaciones: UML, OMT y Booch. * Permite desarrollo multiusuario. * Genera documentación del sistema. * Disponible en múltiples plataformas. | * Soporte a modelos de análisis, ANSI C++, RoseJ y Visual C++ según el documento "Design Patterns: Elemts of Reusable Object – Oriented Software". * Los componentes del modelo se pueden controlar independientemente, lo que permite una gestión y un uso de modelos mas granular. * Soporte para compilación y descompilación de las construcciones más habituales de Java 1.5. * Generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código. |
| Espiral | Se comienza con un pequeño conjunto de requisitos y pasa por cada fase de desarrollo para ese conjunto de requisitos. El equipo de desarrollo agrega la funcionalidad para el requerimiento adicional en espirales cada vez mayores, hasta que la aplicación está lista para la fase de producción. | | * La funcionalidad adicional o los cambios se pueden hacer en una etapa posterior. * La estimación del coste se hace fácil, ya que la construcción del prototipo se hace en pequeños fragmentos. * El desarrollo continuo o repetido ayuda en la gestión de riesgos. * El desarrollo es rápido y las características se añaden de forma sistemática. * Siempre hay espacio para atender los comentarios de los clientes. * Riesgo de no cumplir con la planificación o el presupuesto. * Funciona mejor para proyectos grandes, aunque en estos también requiera de una estricta evaluación de riesgos. * Para su buen funcionamiento, el protocolo del modelo en espiral debe ser seguido estrictamente. * Se genera más documentación al tener fases intermedias. * No es aconsejable para proyectos pequeños, la ratio coste beneficio no es rentable. | **Planificación**  Incluye la estimación del coste, el calendario y los recursos para la iteración.  Implica también la comprensión de los requisitos del sistema para la comunicación continua entre el analista de requerimientos y el cliente.  **Análisis del riesgo**  La identificación de los riesgos potenciales se realiza mientras se planifica y finaliza la estrategia de mitigación de riesgos.  **Ingeniería**  Incluye la codificación, pruebas y el despliegue del software.  **Evaluación**  Evaluación del software por parte del cliente.  Además, incluye la identificación y el seguimiento de riesgos tales como los retrasos en los plazos y los sobrecostes. |
|  |  | |  |  |
| Modelo de Desarrollo de Software | ***Clasificación*** | | ***Uso*** | ***Ejemplo*** |
| Modelo Prototipo | **Modelo de Prototipos rápido:** Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y prescinde del prototipo cuando el próximo diseño es desarrollado mediante un nuevo prototipo.  **Modelo de Prototipos reutilizable:** También conocido como "Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real. Mayormente es utilizado en el desarrollo de software, si bien determinados productos de hardware pueden hacer uso del prototipo como la base del diseño de moldes en la fabricación con plásticos o en el diseño de carrocerías de automóviles.  **Modelo de Prototipos Modular:** También conocido como Prototipado Incremental (Incremental prototyping); se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresa.  **Modelo de Prototipos Horizontal:** El prototipo cubre un amplio número de aspectos y funciones, pero la mayoría no son operativas. Resulta muy útil para evaluar el alcance del producto, pero no su uso real.  **Modelo de Prototipos Vertical:** El prototipo cubre sólo un pequeño número de funciones operativas. Resulta muy útil para evaluar el uso real sobre una pequeña parte del producto.  **Modelo de Prototipos de Baja-fidelidad:** El prototipo se implementa con papel y lápiz, emulando la función del producto real sin mostrar el aspecto real del mismo. Resulta muy útil para realizar tests baratos.  **Modelo de Prototipos de Alta-fidelidad:** El prototipo se implementa de la forma más cercana posible al diseño real en términos de aspecto, impresiones, interacción y tiempo. | | El modelo de prototipos, también conocido como modelo de desarrollo evolutivo, es usado principalmente en proyectos de desarrollo de software. Este modelo se utiliza para dar al usuario una vista preliminar de lo que será el sistema. (Desarrollo de Aplicaciones Móviles & de Escritorio) | Página web en diversas etapas con diversas características antes de su publicación. |
| Cascada | | El modelo en cascada de cinco niveles, basado en las propuestas de Winston W. Royce, divide los procesos de desarrollo en las siguientes fases de proyecto: análisis, diseño, implementación, verificación y mantenimiento. | Desarrollo de Software (Aplicaciones Web, de Escritorio, etc.) | El modelo de cascada se utilizó para desarrollar aplicaciones empresariales como sistemas de gestión de relaciones con los clientes (CRM), sistemas de gestión de recursos humanos (HRMS), sistemas de gestión de la cadena de suministro, sistemas de gestión de inventarios, sistemas de puntos de venta (POS) para cadenas minoristas, etc. |
| Dientes de Sierra | | •Al encontrarse errores luego de realizar las pruebas se pierde tiempo y dinero, ya que cada prueba se realiza luego de haber terminado la implementación.  •Las pruebas de cada fase ayudaran a corregir posibles errores sin tener que esperar a que sean rectificados en la etapa final del proceso.  •Con las pruebas unitarias y de integración se consigue obtener exactitud en los programas. | Desarrollo de Software (Páginas Web) | MODELO DE VIDA DEL SOFTWARE: MODELO DE CICLO DE VIDA DEL SOFTWARE |
| Rational Rose | | * Soporte para enterprise Java Beans 2.0. * Funciones de análisis de calidad de código. * Complemento de modelado Web que incluye funciones de visualización, modelado y herramientas para desarrollar aplicaciones Web. * Modelado en UML para diseñar bases de datos, que integra los requisitos de datos y aplicaciones mediante diseños lógicos y analíticos. | * Creación de definiciones de tipo de documentos DTD en XML. * Integración con otras herramientas de desarrollo de IBM Racional. * Integración con cualquier sistema de control de versiones compatibles con SSC, como IBM Rational ClearCase. * Posibilidad de publicar en las Web modelos e informes para mejorar la comunicación entre los miembros del equipo. | Desarrollo de diagramas UML para describir las actividades de una empresa, en uno o todas sus divisiones. |
| Espiral | | Se planificarán los siguientes pasos y se comienza un nuevo ciclo de la espiral. La espiral tiene una forma de caracola y se dice que mantiene dos dimensiones, la radial y la angular:  **Angular:** Indica el avance del proyecto del software dentro de un ciclo.  **Radial:** Indica el aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando. | El desarrollo o modelo en espiral es un enfoque de desarrollo de software que puede ser considerado como una respuesta a los inconvenientes del desarrollo en cascada. El modelo en espiral describe el ciclo de vida de un software por medio de espirales, que se repiten hasta que se puede entregar el producto terminado. | Qué es el desarrollo en Espiral? | Deloitte España |
|  | |  |  |  |
|  | |  |  |  |