Modelos de procesos de software

Diseño de sistemas de informacion

elkin de jesus ramirez jimenez

2017

1. **¿Que es un proceso de software?**

Las metas de la ingeniería de software es construir productos de software, o mejorar los existentes; en ingeniería de procesos, la meta es desarrollar o mejorar procesos.

Un proceso de desarrollo de software es un conjunto de personas, estructuras de organización, reglas, políticas, actividades y sus procedimientos, componentes de software, metodologías, y herramientas utilizadas o creadas específicamente para definir, desarrollar, ofrecer un servicio, innovar y extender un producto de software.

* 1. Un proceso de software efectivo habilita a la organización a incrementar su productividad al desarrollar software:
* Permite estandarizar esfuerzos, promover reusó, repetición y consistencia entre proyectos.
* Promueve la oportunidad de introducir mejores prácticas de la industria.
* Permite entender que las herramientas deben ser utilizadas para soportar un proceso.
* Establece la base para una mayor consistencia y mejoras futuras.
  1. Un proceso de software mejora los esfuerzos de mantenimiento y soporte:
* Define cómo manejar los cambios y liberaciones a sistemas de software existentes.
* Define cómo lograr la transición del software a la operación, y cómo ejecutar los esfuerzos de operación y soporte.

1. **Modelos de procesos de software:**
   1. **Modelo cascada:**

Este modelo es el más conocido, está basado en el ciclo convencional de una ingeniería. Para pasar de una fase a otra debe de estar completamente terminada revisas y aprobada por el departamento de desarrollo esto con el fin de llevar un orden estricto funcional de los procesos.

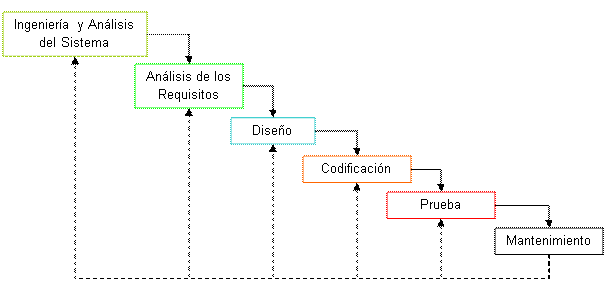


Ilustración 1Modelo Cascada

**2.1.1) Características de modelo cascada**

* Es una visión del proceso de desarrollo de software como una sucesión de etapas que produce productos intermedios.
* Si se cambia el orden de las fases, el producto final será de inferior calidad.
  + 1. **Clasificación:**
* Planificación.
* Desarrollo.
* Mantenimiento.
  + 1. **Etapas:**
* Diseño del Sistema:

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

* Diseño del Programa:

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

* Codificación:

Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.

Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

* Pruebas:

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser puesto

* Implantación:

El software obtenido se pone en producción. Se implantan los niveles software y hardware que componen el proyecto. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto. Es una de las fases finales del proyecto.

Durante la explotación del sistema de software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras. Todo ello se recoge en los Documentos de Cambios.

* 1. **Modelo evolutivo:**

Los modelos evolutivos son modelos iterativos, permiten desarrollar versiones cada vez más completas y complejas, hasta llegar al objetivo final deseado; incluso evolucionar más allá, durante la fase de operación.

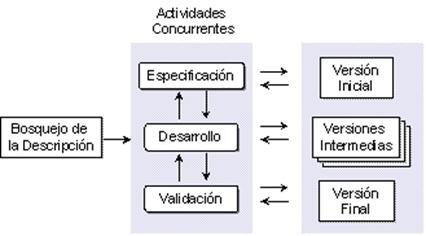


Ilustración 2Especificaciones

**2.2.1) Características:**

* Es evolutivo
* Posee un enfoque evolutivo para la creación de software
* Comienza con la identificación de las clases más importantes
* Examina los datos que se van a manejar
* Permite la reutilización del software
* El ensamblaje de los componentes reduce el 70 del 100% del tiempo del ciclo del desarrollo del software y un 84 del 100% del costo del proyecto.

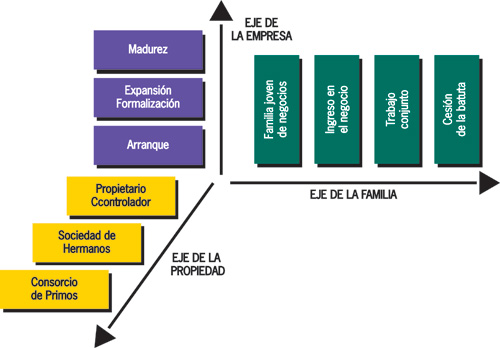
****

Ilustración 3Modelo Evolutivo

**2.2.2) Etapas:**

* PLANEACION

En esta etapa evalúa la función y el rendimiento que se asignaron al Software durante la Ingeniería del Sistema de Computadora para establecer un ámbito de proyecto que no sea ambiguo, e incomprensible.

* ANÁLISIS DE RIESGOS

En esta etapa l analista se encarga de analizar los riesgos que el software a crear estará expuesto y así encontrar la manera de corregirlos.

* CONSTRUCCIÓN Y ADAPTACIÓN DE LA INGENIERÍA

En esta etapa se construye el software, se prueba si no tiene algún problema o para detectar errores, se instala, y luego se le brinda soporte al cliente.

* VALUACIÓN DEL CLIENTE

El cliente tiene la tarea de evaluar el software para verificar si este cumple con los requisitos que este proporciono y está en toda la tarea de aprobar o rechazar el software.

1. **Modelo incremental:**

El enfoque incremental de desarrollo se ve como una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema.

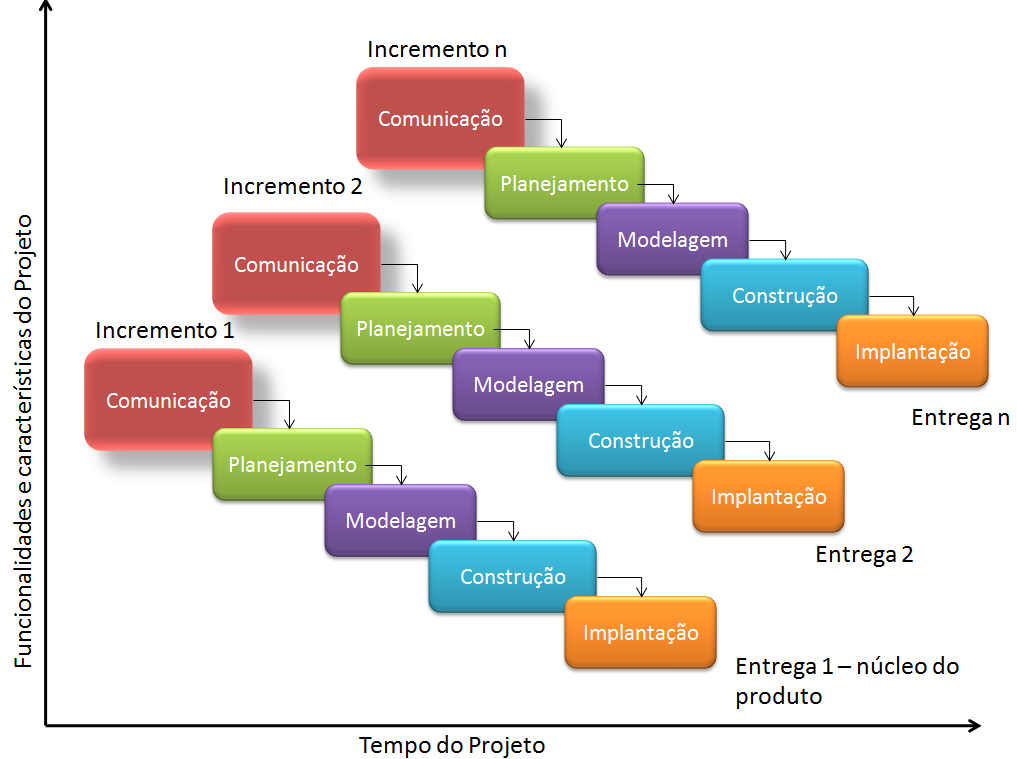


Ilustración 4modelo incremental

**3.0.1) Características:**

* + 1. Se evitan proyectos largos y se entrega "algo de valor" a los usuarios con cierta frecuencia.
    2. El usuario se involucra más.
    3. Difícil de evaluar el costo total.
    4. Difícil de aplicar a los sistemas transaccionales que tienden a ser integrados y a operar como un todo.
    5. Requiere gestores experimentados.
    6. Los errores en los requisitos se detectan tarde.
    7. El resultado puede ser positivo.

**3.0.2) Clasificación:**

* Inicio
* Desarrollo
* Estandarización
* Innovación
* Sustentabilidad

**3.0.3) Etapas:**

* Requerimientos
* Definición de tareas
* Diseño de incrementos
* Desarrollo de incrementos
* Validación de incrementos
* Integración de incrementos
* Entrega del producto

1. **Modelo espiral**



El modelo espiral en el desarrollo de software es un modelo meta del ciclo de vida del software donde es el fuerzo del desarrollo es iterativo, tan pronto culmina un esfuerzo del desarrollo por hay mismo comienza el otro. Además, en cada ejecución del desarrollo se sigue cuatro pasos principales.

**4.0.1) Características:**

* Comunicación con el cliente
* Planificación
* Análisis de riesgos
* Ingeniería
* Construcción y adaptación
* Evaluación del cliente

**4.0.2) Clasificación:**

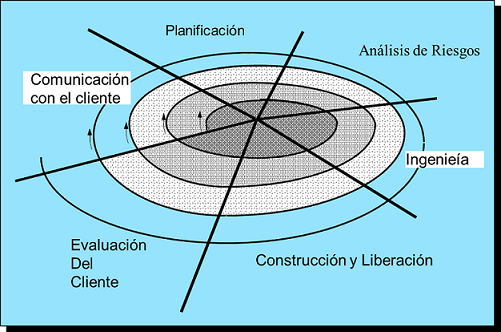


Ilustración 5Modelo espiral etapas

**4.0.3) Etapa:**

* Comunicación con el cliente:

Las tareas requeridas para establecer comunicación entre el desarrollador yel cliente.

* Planificación

Las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.

* Análisis de riesgos

Las tareas requeridas para evaluar riesgos técnicos y de gestión.

* Ingeniería

Las tareas requeridas para construir una o más representaciones de la aplicación.

* Construcción y acción

Las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (por ejemplo: documentación y práctica)

* Evaluación del cliente

Las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

1. **Cuadro comparativo: (BUZZ)**

|  |  |  |
| --- | --- | --- |
| Modelo | Ventajas | desventajas |
| Cascada | •Se tiene todo bien organizado y no se mezclan las fases.  •La planificación es sencilla.  •La calidad del producto resultante es alta. | •Se tarda mucho tiempo en pasar por todo el ciclo.  •Es difícil incorporar nuevas cosas si se quiere actualizar.  •Iteraciones costosas |
| Evolutivo | • La especificación puede desarrollarse de forma creciente.  •Los usuarios y desarrolladores logran un mejor entendimiento del sistema. Esto se refleja en una mejora de la calidad del software.  • Es más efectivo que el modelo de cascada, ya que cumple con las necesidades inmediatas del cliente. | •Proceso no Visible: Los administradores necesitan entregas para medir el progreso. Si el sistema se necesita desarrollar rápido, no es efectivo producir documentos que reflejen cada versión del sistema.  •Sistemas pobremente estructurados: Los cambios continuos pueden ser perjudiciales para la estructura del software haciendo costoso el mantenimiento.  •Se requieren técnicas y herramientas: Para el rápido desarrollo se necesitan herramientas que pueden ser incompatibles con otras o que poca gente sabe utilizar. |
| Incremental | •Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.  •También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del software | •Requiere de mucha planeación, tanto administrativa como técnica.  •Requiere de metas claras para conocer el estado del proyecto |
| Espiral | •El modelo en espiral permite a quien desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto | •Tiene una elevada complejidad.  •Es un modelo costoso.  •Genera mucho tiempo en el desarrollo del sistema |

1. **Bibliografía y cibergrafía**

**Modelo cascada**

Academia. (s.f.). Obtenido de http://www.academia.edu/5130339/MODELO\_CARACTERISTICAS\_VENTAJAS\_DESVENTAJAS\_CASCADA

**Modelo evolutivo**

blogger. (s.f.). Obtenido de http://jorgetrejos.blogspot.com.co/2010/08/modelo-evolutivo.html

blogger. (s.f.). Obtenido de http://gproyectos-s4b.blogspot.com.co/

blogger. (s.f.). Obtenido de http://gproyectos-s4b.blogspot.com.co/

Blogger. (s.f.). Obtenido de http://ciclosdevida1.blogspot.com.co/2008/09/ciclos-de-vida-clasificacin.html

**Proceso de software**

BUZZ, S. (s.f.). Obtenido de https://sg.com.mx/revista/1/procesos-software#.WaGl7SgjHIV

**Modelo espiral**

EcuRed. (s.f.). Obtenido de https://www.ecured.cu/Modelo\_Espiral

monografias. (s.f.). Obtenido de http://www.monografias.com/trabajos108/modelos-del-proceso-del-software/modelos-del-proceso-del-software.shtml

software, P. (s.f.). Obtenido de https://procesosoftware.wikispaces.com/Modelo+Incremental