**ANEXO**

**“PROCESOS DE SOFTWARE”**

PROCESOS DE SOFTWARE

El **proceso de ingeniería de software** es un conjunto de etapas ordenadas con la intención de lograr la obtención de un producto de software de calidad. En el proceso de desarrollo de software las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo. Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". Evolución.

A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases:

**- concepción:** define le alcance del proyecto y desarrolla un caso de negocio

**- elaboración:** un plan del proyecto, especifica las características y fundamenta la arquitectura

**- construcción:** crea el producto

**- transición**: transición transfiere el producto a los usuarios.

Las actividades varían dependiendo de la organización y del tipo de sistema a desarrollarse.

**Modelos del proceso del software**:

* Modelo lineal
* Modelo de construcción de prototipos
* Modelo de desarrollo rápido de aplicaciones (DRA)
* Modelos evolutivos: incremental, espiral, de desarrollo concurrente
* Modelos de métodos formales
* Técnicas de cuarta generación

**2. MODELO LINEAL-SECUENCIAL**

Llamado en algunos casos “modelo en cascada”, el modelo lineal, es el modelo más antiguo y extensamente utilizado. Se conoce como el ciclo de vida básico. Sugiere un enfoque sistemático, secuencial para el desarrollo del software, que comienza con un nivel de sistemas y progresa con el análisis, diseño, codificación y pruebas.

Análisis

Diseño

Desarrollo

Pruebas

* **Análisis de los requerimientos del software:** es la fase en la cual se reúnen todos los requisitos que debe cumplir el software. En esta etapa es fundamental la presencia del cliente que documenta y repasa dichos requisitos.
* **Diseño:** es una etapa dirigida hacia la estructura de datos, la arquitectura del software, las representaciones de la interfaz y el detalle procedimental (algoritmo). En forma general se hace un esbozo de lo solicitado y se documenta haciéndose parte del software.
* **Desarrollo:** es la etapa en la cual se traduce el diseño para que sea comprensible por la máquina. Esta etapa va a depender estrechamente de lo detallado del diseño.
* **Pruebas:** esta etapa se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en la detección de errores.
* **Mantenimiento:** debido a que el programa puede tener errores, puede no ser del completo agrado del cliente o puede necesitar, eventualmente acoplarse a los cambios en su entorno. Esto quiere decir que no se rehace el programa, sino que sobre la base de uno ya existente se realizan algunos cambios.

**2.1. Inconvenientes:**

* Los proyectos reales rara vez siguen el flujo secuencial
* Es difícil establecer explícitamente al principio todos los requisitos
* El cliente debe ser paciente
* Se producen estados de bloqueo

**3. MODELO DE CONSTRUCCIÓN DE PROTOTIPOS**

Comienza con una recolección inicial de requisitos para pasar a un diseño rápido y finalmente a la construcción de un prototipo de la solución.

El prototipo se utiliza para precisar los requisitos del software y así evitar inconvenientes como.

El cliente cree que el prototipo es una primera versión funcional del Sistema.



**3.1. Ventajas**

* Ayuda a identificar los requisitos
* Agrada tanto a los clientes como a los Desarrolladores

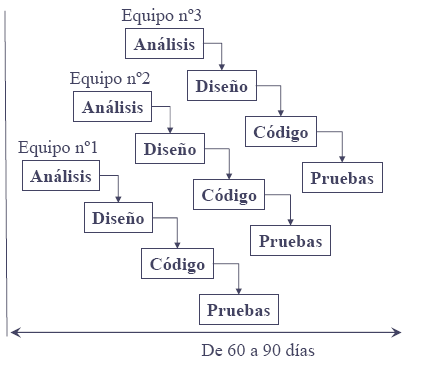
**3.2. Inconvenientes**

* El cliente considera al prototipo como el producto final, listo para usar.
* La calidad del software o la factibilidad de mantenimiento no se tienen en cuenta
* El desarrollador a menudo hace compromisos de implementación

**4. EL MODELO DRA**

“Desarrollo rápido de aplicaciones”. Es un modelo lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto.

* Modelado de Gestión
* Modelado de datos
* Modelado de procesos
* Generación de aplicaciones
* Pruebas y entregas



* 1. **Inconvenientes**
* Para proyectos grandes requiere recursos humanos suficientes
* Los clientes y desarrolladores deben estar comprometidos en las rápidas actividades
* Si el sistema no se puede modularizar será problemático
* No es adecuado con riesgos técnicos altos

1. **MODELOS EVOLUTIVOS DEL PROCESO**

Modelos flexibles que permiten la modificación del sistema durante su proceso de desarrollo.

Procesos iterativos que permiten a los desarrolladores construir versiones del software cada vez más completas

**Ejemplos:**

**Modelo Incremental.**

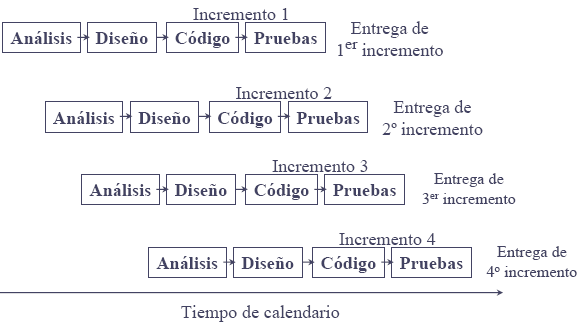
**Modelo Espiral.**

**Modelo Espiral WINWIN.**

**Modelo de Desarrollo Concurrente**

1. **EL MODELO INCREMENTAL**

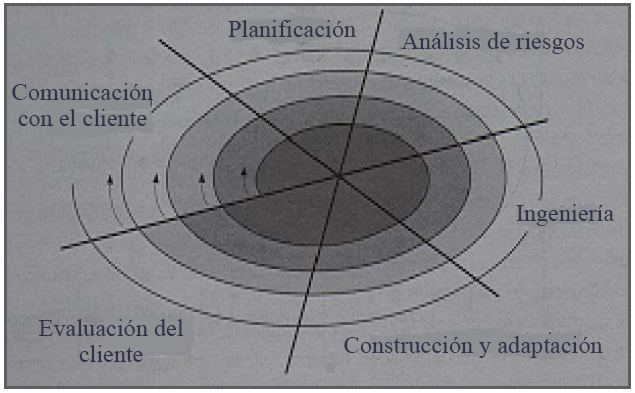
* En este modelo se desarrolla el sistema por partes, Incrementándolas y juntándolas después. Cada incremento agrega funcionalidad adicional o mejorada sobre el sistema (versiones)
* El primer incremento a menudo es un producto esencial (núcleo)
* A partir de la evaluación se planea el siguiente incremento y así sucesivamente
* Los errores producidos en un incremento son solucionados para el próximo incremento.
* Es interactivo por naturaleza
* Es útil cuando el personal no es suficiente para la implementación completa



* 1. **Ventajas**
* Se puede financiar el proyecto por partes
* Apropiado para proyectos grandes de larga duración
* No se necesita tanto personal al principio como para una implementación completa
* Los clientes no tienen que esperar hasta que el sistema se entregue completamente para comenzar a hacer uso de él.
* Los clientes pueden usar los incrementos iniciales como prototipo para precisar los requerimientos posteriores del sistema.
* Minimización del riesgo de falla en el proyecto porque los errores se van corrigiendo progresivamente.
* Es menos arriesgado construir un sistema pequeño que uno grande.
  1. **Inconvenientes**
* Se necesitan pruebas de regresión
* Pueden aumentar el coste debido a las pruebas
* Adaptación de los requisitos del cliente para lograr incrementos pequeños (no más de 20.000 líneas de código) que añadan funcionalidad al sistema.

1. **EL MODELO EN ESPIRAL**

* Es un modelo de software evolutivo que Tiene la naturaleza interactiva de construcción de prototipos y los aspectos controlados y sistemáticos del modelo lineal secuencial
* Utilización de *ciclos* en lugar de sucesión de actividades.
* Se construyen sucesivas versiones del software cada vez más complejas.
* Se eligen múltiples alternativas y se coge la mejor.
* Tan pronto se termina un esfuerzo de desarrollo, otro comienza.
* Cada vuelta de la espiral se resuelve un conjunto particular de problemas del cliente.
* Después de cada vuelta realizamos un prototipo.
* Durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo
* Durante las últimas iteraciones, se producen versiones cada vez más completas de ingeniería de sistemas



* 1. **Tareas:**
* **Comunicación con el cliente:** Para establecer comunicación entre el desarrollador y el cliente.
* **Planificación:** Para definir los recursos, el tiempo y otras informaciones relacionadas con el proyecto.
* **Análisis de riesgos:** Para evaluar riegos técnicos y operativos.
* Ingeniería: Para construir una o más representaciones de la aplicación.
* **Construcción y adaptación:** Para construir, probar, instalar y proporcionar soporte al usuario
* **Evaluación del cliente:** Para obtener la reacción del cliente según la evaluación de las representaciones del software
  1. **Ventajas:**
* Facilita el desarrollo rápido de versiones incrementales de software.
* Se utiliza cuando no se tiene claro los requerimientos.

1. **EL MODELO DE COMPONENTES**

* El modelo utiliza el marco de trabajo técnico del paradigma orientado a objetos
* Incorpora muchas características del modelo en espiral
* La actividad de ingeniería comienza con la identificación de clases candidatas

Según estudios realizados este modelo:

* Reduce el tiempo de desarrollo en un 70%
* Reduce el costo del proyecto en un 84%

**6. MODELO AGILE**

El Desarrollo ágil de Software es un paradigma de las Metodologias De Desarrollo basado en procesos ágiles. Los procesos ágiles de desarrollo de software, conocidos anteriormente como metodologías livianas, intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados.

El proceso ágil usa un enfoque basado en el Valor para construir software, colaborando con el cliente e incoporando los cambios continuamente.

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.

El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas. Cada iteraciones del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, pero la meta es tener un demo (sin errores) al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos Agiles enfatizan las comunicaciones cara a cara a través de la documentación. La mayoría de los equipos Agiles están localizados en una simple oficina abierta. La oficina debe incluir revisores, diseñadores de iteración, escritores de documentación y ayuda y directores de proyecto.

**7. PROCESO UNIFICADO DE SOFTWARE**

Proceso unificado de software lo podemos definir como conjunto de actividades necesarias para transformar los requisitos de los futuros usuarios en un sistema de software. Sin embargo el proceso es además un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas, diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.

Los verdaderos aspectos que definen el Proceso Unificado

* El Proceso Unificado está dirigido por casos de uso
* El Proceso Unificado está centrado en la arquitectura
* El Proceso Unificado es iterativo e incremental

Yo noto una diferencia respecto a los demás modelos en la complejidad de este proceso, con el hecho de trabajar con casos de uso, los cuales son verdaderamente útiles en todo el desarrollo del sistema como en el diseño, implementación, y prueba del sistema.

También el hecho de que se centre en la arquitectura, teniendo en cuenta parámetros previos como por ejemplo la plataforma en la que el producto tiene que funcionar, y también lo ventajoso de ser un modelo de desarrollo iterativo e incremental.

Con respecto a nuestro trabajo lo asociaría como el modelo que podríamos utilizar para el desarrollo del producto, para conocer de forma práctica todo este proceso que demuestra ser muy completo y prometedor de aplicaciones exitosas.