



Sistemas de Información

DOCENTE: CRISTIAN MATAMALA GOMEZ

CLASE - 06 DE JULIO DEL 2021

Agenda

- Metodologías de Desarrollo de Sistemas de Información.

Metodologías de Desarrollo de Sistemas de Información

Video Introductorio

<https://www.youtube.com/watch?v=r0rV48EXlmo>



Metodología para el Desarrollo de SI

Las metodologías son sistemas completos de técnicas que incluyen procedimientos paso a paso, productos resultante, funciones, herramientas y normas de calidad para la terminación del ciclo de vida completo del desarrollo de sistemas.

Metodología para el Desarrollo de SI

Una Metodología para el Desarrollo de SI es un conjunto de actividades llevadas a cabo para desarrollar y poner en marcha un Sistema de Información.

Objetivos y Tipos de Metodologías

- Objetivos de las metodologías
 - Definir actividades a llevar a cabo en un proyecto de S.I.
 - Unificar criterios en la organización para el desarrollo de S.I.
 - Proporcionar puntos de control y revisión.
- Tipos de Metodologías
 - Estructurada
 - Evolutiva-Incremental
 - Prototipos
 - Orientada a objetos

Miembros de un proyecto de Sistemas

- Líder (Gerencia el proyecto).
- Analista (recoge información inicial y define requerimientos).
- Diseñador de S.I.
- Diseñador de Bases de Datos (B.D.).
- Programador (Codifica/Prueba).
- Usuario directo. (Expresa necesidades).

Revisaremos

- Planificación de Proyectos
- Justificación
- Control de Proyectos
- Estudio de Factibilidad
- Análisis
- Diseño
- Implantación
- Actualización

Planificación de Proyectos

- Permite saber qué se deberá hacer y quien lo va hacer. Tiempo estimado de terminación del proyecto (aproximadamente).
- Pone en evidencia los obstáculos relevantes del proyecto, con el fin de tomar las precauciones necesarias.
- Establece marco de referencia que permite trabajar eficientemente y sin desperdicio de recursos.
- Permite definir la metodología de desarrollo a seguir.
- Herramientas para la planificación (Cronograma de Actividades, Software de Planificación [Project]).

Justificación del proyecto

- Se establecen las bases para soportar la aprobación.
- Incluye análisis costo/beneficio.
- Verifica:
 - Definición correcta de objetivos del proyecto.
 - Enunciación correcta de prioridades.
 - Optimización de beneficios
 - Razones para Proponer proyectos:
 - Resolver un problema /necesidad.
 - Aprovechar una oportunidad.
 - Brindar soluciones a directivos.

Justificación del proyecto

- Razones para iniciar proyectos
 - Mayor capacidad (velocidad, memoria, recursos)
 - Control.
 - Reducción de costos.
 - Alcanzar ventajas competitivas.

Control de Proyectos

- **Tareas del líder de Proyectos**
 - Prepara y ejecutar planes de acción.
 - Dirigir reuniones para identificar y resolver problemas.
 - Elaborar y presentar reportes de progresos.
- **Ventajas de control de proyecto**
 - Permite reasignar personas con poca carga.
 - Permite intercambiar personal de actividades no críticas a críticas
- **Proyecto bajo control**
 - Cada persona sabe lo que tiene que hacer y cuándo debe hacerlo.
 - Nadie está esperando que las cosas ocurran.
 - No hay problemas escondidos.
 - El líder sabe lo que se ha hecho y lo que no.

Control de Proyectos

- Para mantener un proyecto bajo control
- Preparar y seguir planes de acción.
- Realizar reuniones para detectar y corregir problemas.
- Delegar eficientemente.
- Medir el tiempo que realmente hace falta.
- Reconocer los síntomas del fracaso.

Estudio de factibilidad

- Determina si es posible o no ofrecer solución automatizada a los problemas/situaciones actuales.
- Representa el primer paso a cumplirse dentro del ciclo de desarrollo.
- Brinda información muy amplia acerca de la unidad a quien se la va a desarrollar el S.I.
- Abarca la factibilidad:
 - Técnica (existe tecnología para realizar el S.I).
 - Operativa (habrá resistencia al cambio).
 - Económica (relación costo/beneficio).

Estudio de Factibilidad

- Beneficios
 - Ahorros funcionales
 - Reducción de costos de operación (tiempo, dinero).
 - Beneficios tangibles
 - Aumento de productividad
 - Mejor uso de los activos
 - Mejor control
 - Beneficios intangibles
 - Optimización o simplificación de procedimientos.
 - Mayor entusiasmo en los trabajadores.
 - Imagen de la organización.
 - Mejora en la precisión de las operaciones

Estudio de factibilidad

Costos:

- Construcción del sistema
 - Sueldos de los miembros del proyecto.
 - Adiestramiento (de ser necesario).
- Operación del sistema
 - Software
 - Hardware
 - Mantenimiento.

Análisis

- Amplía resultados del estudio de factibilidad
- Define **QUÉ** va a hacer el nuevo sistema
- Herramientas
 - Técnicas de recolección de información (entrevistas, cuestionarios)
 - Descripciones de procesos y procedimientos
 - Diagramas de flujo de datos (herramienta gráfica que se emplea para describir y analizar el movimiento de datos a través de un sistema)
 - Diagramas de flujo de procesos (representa el modelaje físico de un sistema, quién y cómo hace las cosas)
 - Diccionario de datos (datos de los datos del sistema, catálogo de los elementos de un sistema)

Análisis Preliminar

- Para el éxito de un desarrollo de un SI es esencial una comprensión total de los requisitos.
- No importa lo bien diseñado o codificado que esté un programa, si no se ha analizado correctamente, pues defraudará al usuario y frustrará al desarrollado.

Análisis

- Definición de objetivos específicos del sistema.
- Identificación de usuarios.
- Identificación de procedimientos propuestos
- Elaboración de modelo del sistema actual (lógico y físico).
- Recopilación de reportes del sistema actual.
- Elaboración del modelo del sistema propuesto (lógico y físico).
- Mostrar beneficios potenciales del sistema propuesto.
- Refinación del plan de trabajo y costos.
- Elaboración de planificación del proyecto.
- Elaboración del diccionario de datos.

Análisis de Requisitos

- El análisis de los requisitos es una tarea de ingeniería de software que cubre un hueco entre la definición del software a nivel sistema y el diseño del software.
- El análisis de requisitos permite al ingeniero de sistemas especificar la función y el rendimiento del software, indica la interfaz del software con otros elementos del sistema y establece que debe cumplir el software

Diseño

- Genera soluciones a requerimientos planteados.
- Define CÓMO lo va hacer el nuevo sistema.
- Herramientas:
 - Lenguaje de Modelado Unificado (UML), diagramas entidad relación, diagrama estructurado, normalización, diccionario de datos, etc.).
- Base de datos (colección integrada de archivos accesibles por múltiples aplicaciones).

Diseño de Salidas

- Diseño de salidas
- Deben satisfacer objetivos planteados.
- Se deben adaptar al usuario.
- Debe de proveer la cantidad adecuada de información.
- Se debe proporcionar el método apropiado para la salida
(reportes impresos, en pantalla, archivos y en discos).
- La salida debe de ser oportuna y disponible.

Programación/Codificación

- Consiste en traducir el diseño en instrucciones que la computadora pueda interpretar.
- Es la generación del código fuente y código objeto de la aplicación, de acuerdo a los resultados obtenidos en el diseño.
- Actividades a cumplir
 - Codificación.
 - Compilación (corregir sintaxis).
 - Depuración (corregir errores de los programas).

Implantación

- Incluye todas las actividades para poner un sistema en producción (entregar al usuario).
- Actividades
 - Prueba.
 - Conversión.
 - Instalación de hardware y software.
 - Adiestramiento.
 - Documentación.
 - Entrega al usuario.

Mantenimiento

- Modificar, corregir o mejorar los sistemas existentes.
- Tipos de Mantenimiento:
 - Correctivo (elimina errores).
 - Perfectivo (añade nuevas funciones).
 - Adaptivo (modifica acciones).
 - Preventivo (previene errores).
- Parches: modificaciones menores.
- Importancia de un mantenimiento:
 - Si no hay apoyo continuo, el sistema puede dejar de funcionar.
 - Un soporte continuo permite a los usuarios el uso adecuado del sistema.
 - Permite realizar ajustes necesarios para que aún cuando el ambiente cambie, se pueda hacer uso eficiente de los recursos del sistema.

Mantenimiento

- Dificultades encontradas:
 - Documentación inadecuada, obsoleta o inexistente.
 - Componentes complejos.
 - Componentes mal estructurados.
 - Poca familiaridad de las aplicaciones.
 - Presión del tiempo.
 - Falta de comunicación y participación de los usuarios.
 - Gran cantidad de requerimientos.
 - Gran cantidad de parches.

Resumen

- Metodologías de Desarrollo de Sistemas de Información.