

INTRODUCCIÓN

domingo, 4 de agosto de 2019

06:22 p. m.

► Ingeniería de Software

► Consideraciones

- Tareas/Presentaciones | SIAEFI - Único medio de entrega (2 weeks)
 - |
 - |
 - |
 - |
 - |
 - |
 - |
 - |
- Cumplir con el 80% → Punto Extra de asistencia sobre Final
- Acerca de las Tareas |
 - ① Subir tareas con formato PDF.
 - ② No se aceptan tareas tarde

► Evaluación

• Tareas	- 50 %	* Según aplique * Un punto sobre alguna sección ↳ DIE - Conferencias ↳ Foto de nosotros durante la conferencia - Comentario de.
• Exposición/*	- 20%	
• Ejercicio de Close		
• 1º parcial		
• 2º parcial *		
• 3º parcial *		
• Asistencia	10%	
		110 %

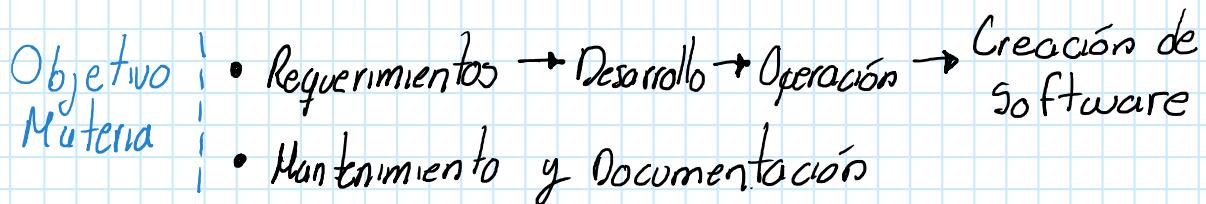
1_Ingeniería de Software

Tuesday, August 6, 2019 7:22 PM

* IEEE

- Aplicación de un enfoque sistemático (ordenado), disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es: **LA APLICACIÓN DE LA INGENIERÍA EN EL ÁREA DE SOFTWARE**

* Fritz Bauer (1972)



► Historia

World Wide Web	{	ENIAC - 1946
		• Disco Duro - 5 mb

► Crisis del Software

1^o Fase - Los albores (1945-1955)

- Programar no es una tarea diferenciada del diseño de una máquina

- Uso de lenguaje máquina y ensamblador

2^o Fase - El florecimiento (1955-1965)

- Aparecen multitud de lenguajes
- Es posible hacer todo

3º Fase - La Crisis (1965-1970)

- Desarrollo inalcanzable de grandes programas.
- Ineficiencia, errores, coste impredecible.
- Nada es posible.

4º Fase - Innovación Conceptual (1970-1980)

- Fundamentos de programación.
- Verificación de la programación.
- Metodologías de diseño.

5º Fase - El diseño del problema (1980-d?)

- Entornos de programación.
- Especificación formal
- Programación automática

→ Crisis de Software

- No se obtenía los resultados además de un gran costo y poca flexibilidad
- Muchos desarrollos sobrepasan el presupuesto y tiempo estimados

// Pensamiento

- Todavía se piensa que la programación debe considerar se un arte: "más creatividad que disciplina".
 - * No se siguen procesos
 - * No hay interés en tener claridad sobre los requerimientos del cliente.
 - "Se oye el problema y se piensa en la solución."

→ "Se oye el problema y se piensa en la solución."

- El programador no tuvo una educación formal, aprendió de la experiencia

// Forma de trabajo

- Enfoque equivocado en el proceso de desarrollo

1/4 - Análisis y Diseño

3/4 - Proyecto y parte de desarrollo

// Nueva perspectiva

- Software como producto y no como programa
 - Ser fiable
 - Ser entregado a tiempo al usuario.
 - Ser portable
- Ser fácil de modificar
- Ser entregado conforme al costo establecidos al usuario.
- Ser predecible en su funcionamiento.

2_P

3_TiposSw_Ciclo

Tuesday, August 20, 2019 8:10 PM

► Tipos de Software (Propósitos)

► Software de aplicación

- Sirve para ayudar al usuario
- Resultado de la programación de software
- Paquetería de office, Adobe Reader.

► Software de sistema

- Diseñado para operar en el hardware del equipo y proporcionar/mantener una plataforma para ejecutar software de aplicación
- Sistemas Operativos

► Software de programación

- Programa o aplicación que los desarrolladores de software
- Lenguajes de programación

► Ciclos Básicos de Desarrollo y de vida del Software

Rueda Aguilar Karla Vanessa
Díaz Montaño Bárbara María Fernanda

- Es un marco de referencia que contiene los procesos, actividades y tareas involucradas en el desarrollo, explotación y mantenimiento de un producto de software.
- Abarcando la vida del sistema desde la definición hasta la finalización de uso.

Abarcando la vida del sistema desde la verificación hasta la finalización de uso.

- Busca validar el desarrollo de la aplicación, es decir, para garantizar que cumpla los requisitos y verificar los procedimientos los procedimientos de desarrollo, buscando asegurar que los métodos utilizados son apropiados.

► Valor del ciclo de vida

- Entendible
- Visibilidad: Grado en que las actividades del proceso proporcionan resultados.
- Soportable: Por herramientas CASE (Computer Aided Software Engineering)
- Aceptable: Grado en que los desarrolladores aceptan y usan el proceso
- Fiableidad: Capacidad de evitar o detectar errores antes de que sean defectos
- Robustez: Continuidad del proceso a pesar de los problemas
- Mantenible: Capacidad de evolución para adaptarse

► Ciclo más básico (Software)

→ Planificación y análisis

- Análisis de requisitos: Los clientes pueden tener una idea general de tipo de producto que necesitan.

Ideas generales de tipo de proyecto que necesitan
(No aporta nada de cómo debe ser la app)

- Fijar el alcance del proyecto de desarrollo y ponerlo en un documento
- * Requisito → Particular * Requerimiento → General

→ Fases de desarrollo, pruebas y documentación

- Desarrollo y programación del software
- Prueba de software
 - Asegura que los errores sean detectados en fases tempranas y ser corregidos
- La documentación
 - Se realiza durante la programación del código
 - Ayuda a mantener y mejorar el programa

→ Despliegue y mantenimiento

- Involucrando instalación, personalización, tests e incluso períodos de evaluación de rendimiento
- Cuando el software ya ha sido probado intermedio
- Recomendable: Incluir formación y soporte porque el software es efectivo cuando se usa de forma apropiada

→ Otras del software

→ Retiro del Software

- Se reemplaza por versiones nuevas
- Cese el soporte total o parcial tras cierto tiempo.
- Se deberá terminar las acciones necesarias para conseguir el retiro
 - ∴ Desarrollar y documentar los pasos necesarios para efectuar el retiro

► Modelos simples del Proceso de Desarrollo de Software

Metodología Tradicionales

- Rigidez ante cambios (lento y moderado)
- Clientes interactúan en reuniones
- Grupos de gran tamaño y distribuidos en varios sitios
- Dependencia de la arquitectura mediante modelos
- Poco feedback (Extiende tiempo)
- Normas de estándares de desarrollo
- Procesos controlados
- Seguimiento del plan inicial

Metodologías Ágiles

- Flexibilidad ante los cambios del proyecto
- Clientes son parte del equipo
- Grupos pequeños y en un mismo lugar
- Menor dependencia de la arquitectura
- Acorta tiempo (Mucho feedback)
- Heurística a partir de prácticas
- No son controlados
- Respuesta ante cambios
 - XP - Xtreme programming
 - SCRUM

Estandar

- ① Debe ser reconocido
- ② Debe cumplir con ciertos requisitos.

► Ingeniería de Software

- Abordar el software mediante un proceso ingenieril:
 - Respetar los tiempos y costos estipulados
 - Que sea eficiente y cumpla con las prestaciones requeridas por el usuario
- Para poder producir Software de calidad, se adopte un enfoque sistemático y organizado en su trabajo.

→ Principios Éticos

- 1] Sociedad → Actuar en forma congruente
- 2] Cliente → Conclien los mejores intereses de sus clientes
- 3] Producto → Cumpla con los estándares profesionales más alto posible.
- 4] Jucio → Integridad e independencia
- 5] Administración → Enfoque ético en la administración del desarrollo y mantenimiento software.
- 6] Profesión → Incrementar la integridad y reputación
- 7] Colegas → Apoyar y ser justos.

→ Principios de Software

- ① El software se desarrolla o modifica con intelecto no se manufatura en el sentido clásico.

• El software se desarrolla o modifica con intelecto, no se manufacura en el sentido clásico.

// Hardware se gasta o se vuelve obsoleto.

② El software no se desgasta.

// La mayor parte del software se construye para un uso individualizado.



③ Hay similitudes entre el desarrollo de software y el hardware

- Aunque la industria se mueve hacia la construcción basada en componentes, la mayor parte de software se construye para un uso individualizado
- Los componentes reutilizables han sido para que el ingeniero pueda concentrarse en los elementos verdaderamente innovadores de diseño