

Ingeniería del software

1.1 Ingeniería del software

Ingeniería (Diccionario de la Real Academia Española (DRAE)):

1) Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía.

2) Profesión y ejercicio del ingeniero

Ingeniero (DRAE) Persona que profesa o ejerce la ingeniería

Ingeniería (Real Academia de Ciencias Exactas, Físicas y Naturales de España) Conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones u otras realizaciones provechosas para el hombre.

^{01/11/02} Pero ^{IAGP} no definen el término Ingeniería de Software

1

Ingeniería del software

Ingeniería de Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software (Zelkovitz, 1978).

Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos (Bohem, 1976)

Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener de modo rentable que sea fiable y trabaje en máquinas reales (Bauer, 1972)

La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software (IEEE, 1993)

IAGP

2

Ingeniería del software

El entorno actual de desarrollo de sistemas software viene adoleciendo de:

- retrasos considerables en la planificación
- poca productividad
- elevadas cargas de mantenimiento
- demandas cada vez más desfasadas con las ofertas
- baja calidad y fiabilidad del producto
- dependencia de los realizadores

Se ha denominado comúnmente "**crisis del software**"

01/11/02

IAGP

3

Ingeniería del software

¿Cómo se define la palabra crisis ?

Crisis se define en el diccionario como "un punto decisivo en el curso de algo; momento, etapa, o evento decisivo o crucial". Sin embargo para el software no ha habido ningún punto crucial, sólo una lenta evolución.

La crisis en la industria del software ha permanecido durante muchos años, lo cual parece una contradicción para el término. Lo que si se podría decir es que hay un problema crónico en el desarrollo de software.

01/11/02

IAGP

4

Ingeniería del software

Las nuevas metodologías suponen un enfoque **integral** del problema, abarcando todas las fases, que en su mayoría no se consideraban en los desarrollos tradicionales.

En particular son fundamentales la reducción de costes y plazos, así como la calidad del producto final.

Estas tecnologías constituyen la denominada "Ingeniería del Software", que se puede definir como **"el tratamiento sistemático de todas las fases del ciclo de vida del software"**.

Anteriormente se han visto otras definiciones, pero todas inciden en la importancia de una disciplina de ingeniería para el desarrollo de software.

01/11/02

IAGP

5

Ingeniería del software

1.2 Sistemas de información

“Un conjunto de cosas que ordenadamente relacionadas entre sí contribuyen a un determinado objetivo” [DRAE, 1992]

Elementos

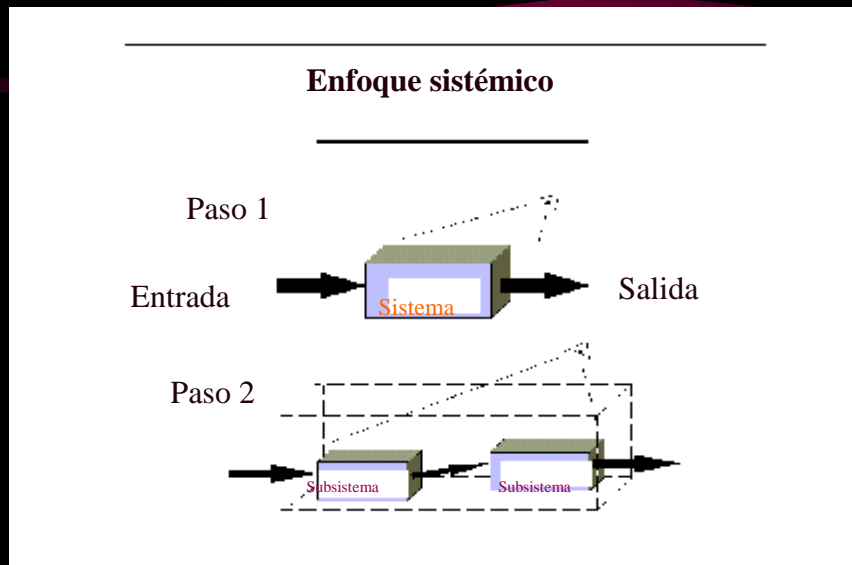
- Los componentes del sistema
- Las relaciones entre ellos
- La estructura del sistema
- El objetivo del sistema
- El entorno del sistema
- Los límites del sistema

01/11/02

IAGP

6

Ingeniería del software



Ingeniería del software

DEFINICION DE SISTEMAS DE INFORMACION

“Un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan, elaboran y distribuyen la información (o parte de ella) necesaria para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) para desempeñar su actividad de acuerdo a su estrategia de negocio”.

Un sistema puede ser tan simple como el indicado, una persona tiene un microordenador y le introduce datos, como por ejemplo las ventas diarias de una empresa. Se produce una entrada por cada venta, declarándose el producto vendido, por ejemplo un yogur, la cantidad de elementos vendidos y el precio de venta unitario. Cada entrada se almacena como un registro de un fichero en el disco. Al finalizar el día se puede obtener un informe de las ventas totales.

Ingeniería del software

ELEMENTOS DE UN SISTEMA DE INFORMACION:

- Los procedimientos y las prácticas habituales de trabajo
- La información
- Las personas o usuarios
- El equipo de soporte

01/11/02

IAGP

9

Ingeniería del software

1.3 Desarrollo histórico de la ingeniería del software

Durante los primeros años de la informática, el software se consideraba como un añadido. La programación era un "arte" para el que no existía metodología, se realizaba sin ninguna planificación.

Toda la programación se desarrollaba a medida para cada aplicación, y en consecuencia tenía muy poca difusión. Habitualmente quien lo escribía era porque lo necesitaba, y era quien lo mantenía.

En una segunda época (a partir de mitad de la década de 1960) se estableció como producto y aparecieron las empresas dedicadas al desarrollo y distribución masiva del mismo.

La tercera era comenzó a mediados de la década de 1970, cuando los sistemas informáticos aumentaron mucho en su complejidad, y nacieron las redes de ordenadores.

01/11/02

IAGP

10

Ingeniería del software

La cuarta era comienza hacia 1990 y se dirige al impacto colectivo de los ordenadores y el software, en todos los entornos. La industria del software tiene un gran peso en la economía mundial. Aparecen las técnicas de redes neuronales, junto con la lógica difusa.

Hoy en día el software tiene un doble papel, es un producto, pero simultáneamente es el vehículo para hacer entrega de un producto.

Como producto permite el uso del hardware, ya sea, por ejemplo un ordenador personal o un teléfono móvil. Como vehículo utilizado para hacer entrega del producto, actúa como base de control, por ejemplo un sistema operativo, o un sistema gestor de redes.

El software hace entrega de lo que se considera como **el producto más importante del siglo veintiuno, la información.**

01/11/02

IAGP

11

Ingeniería del software

La ingeniería del software trata áreas muy diversas de las ciencias de la computación, aplicables a un amplio espectro de campos.

Es habitual que en la práctica profesional no se incluya prácticamente ninguna de las recomendaciones más elementales de la ingeniería del software. Es frecuente que el desarrollo de software se parezca más al descontrol del cuento de «si los programadores fueran albañiles...» que a una idílica y bien organizada factoría de software.

Como ejemplo de que la ingeniería del software es en la actualidad imprescindible, la revista satírica inglesa Private Eye ha dado detalles sobre importantes proyectos que han dado resultados malos.

Destacando los del servicio de ambulancias de Londres, el servicio de sanidad regional de Wessex, la Sociedad para los derechos de autor y el sistema de manejo de equipajes del aeropuerto de Denver.

01/11/02

IAGP

12

Ingeniería del software

1.4 Ciclo de vida

CONCEPTO DE CICLO DE VIDA

“Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software. (IEEE 1074)

Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”. (ISO 12207-1)

01/11/02

IAGP

13

Ingeniería del software

Si se representa la proporción de fallos en función del tiempo, para el hardware se tiene la figura conocida como "curva de bañera". Al principio hay bastantes fallos, una vez corregidos se llega a un nivel estacionario. Sin embargo conforme pasa el tiempo, aparecen de nuevo, por efecto de suciedad, malos tratos, temperaturas extremas y otras causas. El hardware empieza a estropearse.

El software no se estropea. La gráfica de fallos en función del tiempo, tiene forma de caída desde el principio, hasta mantenerse estable por tiempo casi indefinido. El software no es susceptible a los males del entorno que provocan el deterioro del hardware. Los efectos no detectados harán que falle el programa durante las primeras etapas de su vida, sin embargo una vez corregidas, no se producen nuevos errores. Sin embargo, aunque no se estropea, si puede deteriorarse debido a los cambios durante su vida.

01/11/02

IAGP

14

Ingeniería del software

Cuando un componente hardware se estropea, se cambia por otro que actúa como una "pieza de repuesto", mientras que para el software, no es habitual este proceso. Esto implica que el mantenimiento de los programas es complejo.

La mayoría del software se construye a medida, en vez de ensamblar componentes previamente creados. Por contra en el hardware se dispone de todo tipo de circuitos integrados, para fabricar de manera rápida un equipo completo. Los ingenieros de software no disponen de esta comodidad, aunque ya se están dando los primeros pasos en esta dirección, que facilitaría tanto el desarrollo de aplicaciones informáticas.

01/11/02

IAGP

15

Ingeniería del software

El software es un elemento lógico, por lo que tiene características diferentes a las del hardware:

- El software se desarrolla, no se fabrica en el sentido clásico de la palabra.
- Ambas actividades se dirigen a la construcción de un "producto", pero los métodos son diferentes.
- Los costes del software se encuentran en la ingeniería, esto implica que los proyectos no se pueden gestionar como si fueran de fabricación.
- A mediados de la década de 1980, se introdujo el concepto de "fábrica de software", que recomienda el uso de herramientas para el desarrollo automático del software.

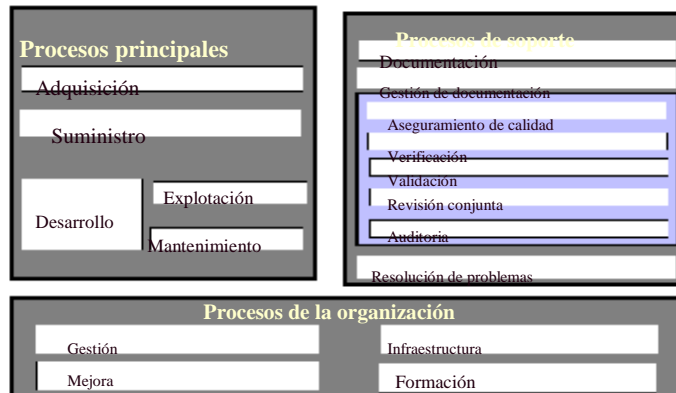
01/11/02

IAGP

16

Ingeniería del software

Procesos del ciclo de vida



Ingeniería del software

PROCESOS PRINCIPALES

Proceso de adquisición

Proceso de suministro

q Proceso de desarrollo (I)

- Análisis de Requisitos del Sistema
- Diseño de la Arquitectura del Sistema
- Análisis de los Requisitos del Software
- Diseño de la Arquitectura del Software
- Diseño Detallado del Software
- Codificación y Prueba del Software

Ingeniería del software

q Proceso de Desarrollo(II)

- Integración del Software
- Prueba del Software
- Integración del Sistema
- Prueba del Sistema
- Instalación del Software
- Soporte del proceso de Aceptación del Software

01/11/02

IAGP

19

Ingeniería del software

Proceso de Explotación

Proceso de Mantenimiento

PROCESOS DE SOPORTE

Proceso de Documentación

Proceso de Gestión de la Configuración

Proceso de Aseguramiento de la Calidad

Proceso de Verificación

Proceso de Validación

Proceso de Revisión Conjunta

01/11/02

IAGP

20

Ingeniería del software

Proceso de Auditoría

Proceso de Resolución de Problemas

PROCESOS GENERALES

Proceso de Gestión

Proceso de Infraestructura

Proceso de Mejora

Proceso de Formación

01/11/02

IAGP

21

Ingeniería del software

1.5 Modelos del ciclo de vida

Aproximación convencional

Se introdujo como una técnica rígida para mejorar la calidad y reducir los costos del desarrollo. Es conocido como "modelo en cascada", porque su filosofía es completar un paso con un alto grado de exactitud, antes de empezar el siguiente.

Los principales problemas se deben a que se comienza estableciendo todos los requisitos del sistema:

01/11/02

IAGP

22

Ingeniería del software

- En muchas ocasiones no es posible disponer de las especificaciones **correctas** desde el primer momento, puede ser difícil para el usuario establecer al inicio todos los requisitos.
- En otras hay cambio de parecer de los usuarios sobre las necesidades reales cuando ya se ha comenzado el proyecto, siendo probables los verdaderos requisitos no se reflejen en el producto final.
- Otro de los problemas de esta aproximación es que los resultados no se ven hasta muy avanzado el proyecto, por lo tanto si ha habido un error, la realización de modificaciones es muy costosa.

01/11/02

IAGP

23

Ingeniería del software

Aproximación prototipo

Es habitual que en un proyecto software, que al inicio no se identifiquen los requisitos detallados de entrada, procesamiento o salida. En otros casos no se está seguro de la eficiencia de un algoritmo, o de la forma en que se ha de implantar la interface hombre-máquina.

En casos así, lo habitual es construir un prototipo, que idealmente sirva como mecanismo para identificar los requisitos del software. Esta aproximación consiste en realizar la fase de definición de requisitos del sistema en base a estos tres factores:

- Un alto grado de iteración
- Un muy alto grado de participación del usuario
- Un uso extensivo de prototipos

01/11/02

IAGP

24

Ingeniería del software

Aproximación evolutiva

En esta aproximación el énfasis está en lograr un sistema flexible y que se pueda expandir de forma que se pueda realizar muy rápidamente una versión modificada del sistema cuando los requisitos cambien.

Se diferencia de la aproximación anterior, en que en esta los requisitos cambian continuamente, lo cual implicaría en el caso previo que las iteraciones no tendrían fin.

01/11/02

IAGP

25

Ingeniería del software

Aproximación incremental

Es un concepto parecido al de desarrollo evolutivo y frecuentemente comprendido en la aproximación del desarrollo evolutivo. Se comienza el desarrollo del sistema para satisfacer un subconjunto de requisitos especificados, las últimas versiones prevén los requisitos que faltan. De esta forma se logra una rápida disponibilidad del sistema, que aunque incompleto, es utilizable y satisface algunas de las necesidades básicas de información.

La diferencia con la aproximación anterior es que en este caso cada versión parte de una previa sin cambios pero con nuevas funciones, mientras que la aproximación evolutiva cada vez se desarrolla una nueva versión de todo el sistema.

01/11/02

IAGP

26

Ingeniería del software

Ventajas de este modelo:

Se evitan proyectos largos y se entrega “algo de valor” a los usuarios con cierta frecuencia

El usuario se involucra más

Mayor retorno de la inversión

Difícil de evaluar el coste total

Difícil de aplicar a sistemas transaccionales que tienden a ser integrados y a operar como un todo

Requiere gestores experimentados

El resultado puede ser muy positivo

01/11/02

IAGP

27

Ingeniería del software

Aproximación espiral

Nace con el objetivo de captar lo mejor de la aproximación convencional y de la de prototipo, añadiendo un nuevo componente, el análisis de riesgos.

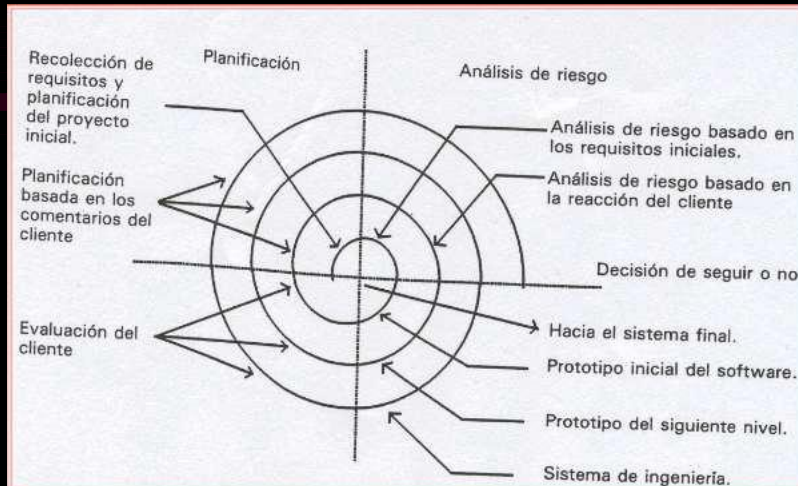
Esquemáticamente se puede ilustrar mediante una espiral, con cuatro cuadrantes que definen actividades.

01/11/02

IAGP

28

Ingeniería del software



01/11/02

IAGP

29

Ingeniería del software

Características:

Definido por Boehm en 1986

Permite acomodar otros modelos

Incorpora objetivos de calidad y gestión de riesgos

Elimina errores y alternativas no atractivas al comienzo

Permite iteraciones, vuelta atrás y finalizaciones rápidas

Es difícil de adaptar a los contratos

Depende de las personas

Difícil de asegurar que las personas involucradas operan en un contexto consistente

01/11/02

IAGP

30

Ingeniería del software

Diferencias con métodos tradicionales:

Existe un **reconocimiento** explícito de las diferentes alternativas para alcanzar los objetivos de un proyecto

La identificación de riesgos asociados con cada una de las alternativas

La división de los proyectos en ciclos

El modelo se adapta a cualquier tipo de actividad

01/11/02

IAGP

31

Ingeniería del software

Aproximación basada en transformaciones

Con la aparición de herramientas CASE, junto con los generadores de código, el ciclo de desarrollo software en cascada ha cambiado a un ciclo de vida basado en transformaciones.

La utilización de estas herramientas afecta a todas las fases del ciclo de vida del software. Este ciclo de vida se puede considerar como una serie de transformaciones. Primero se definen los requisitos del sistema, seguidamente existe un proceso de transformación que hace que la especificación se convierta en un diseño lógico del sistema. Posteriormente, este sufre otro proceso de transformación para lograr un diseño físico, es decir que responda a la tecnología destino.

01/11/02

IAGP

32

Ingeniería del software

La tecnología CASE propone que estos procesos de transformación sean lo más automatizables posible.

Sus ventajas son:

- Posibilidad de comprobación de errores en etapas iniciales.
- Posibilidad de realizar el mantenimiento en el ámbito de especificación.
- Soporte de rastreabilidad de los requisitos.
- Soporte de reusabilidad.
- Potencia la especificación orientada al problema.

01/11/02

IAGP

33

Ingeniería del software

El proceso que siguen las herramientas CASE es:

- Se define el sistema utilizando un lenguaje formal
- La implementación es automática, asistida por el ordenador
- La documentación se genera de forma automática
- El mantenimiento se realiza “por sustitución” no mediante “parches”
- Dificultad en la participación del usuario
- Diseños poco optimizados

01/11/02

IAGP

34

Ingeniería del software

1.6 Metodología de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

Es como un libro de recetas de cocina, en el que se van indicando paso a paso las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Actualmente es imprescindible considerar los riesgos, aunque habitualmente las empresas no han sido concienciadas.

35