CAPITULO 1 PRESSMAN EL PRODUCTO

Dato: Hecho, noción o instrucción que adopta una forma de representación convencional.

# LA EVOLUCIÓN DEL SOFTWARE

El Software es a la vez un producto y el vehículo de entrega de un producto. Como producto puede residir en un teléfono celular o una computadora central donde produce, transforma, adquiere, modifica muestra, transmite información tan compleja como un bit o una presentación multimedia.

Como vehículo para hacer entrega del producto actúa como la base de control de una PC (Sistema Operativo), la comunicación de información (Redes) y creación y control de otros programas (Entornos y herramientas de programación).

El papel del software ha sufrido un cambio importante en la segunda mitad de siglo veinte junto a los avances tecnológicos.

A principios de la era informática El desarrollo de software se realizaba sin ninguna planificación y con gran esfuerzo, el hardware era de propósito general y el software se diseñaba a medida (la misma persona lo escribía, lo ejecutaba y depuraba, no existía casi documentación).

En la segunda era hizo su aparición la multiprogramación y los sistemas multiusuario que produjeron la interacción hombre-maquina que abrieron un nuevo mundo de aplicaciones y sofisticación del hard y soft.

También en esta era aparecieron los sistemas de tiempo real y la primera generación de sistemas de gestión de bases de datos. El soft se estableció como producto y tuvo una amplia distribución y tuvo que ser depurado al aparecer fallas en el mismo con lo cual nació el mantenimiento y junto con los programas personalizados (imposibles de mantener) comenzó una crisis del software.

En la tercera era hicieron aparición los sistemas distribuidos y las redes de área local y global que provocaron gran presión sobre los desarrolladores de software. La aparición del microprocesador produjo muchos productos inteligentes (automóviles, robots, horno a microondas, etc.)siendo él más importante la computadora personal que pronto fue de fácil acceso para él publico.

La cuarta era se alejo de las computadoras individuales hacia el impacto colectivo de computadoras y software. Los entornos centralizados pasaron a ser cliente/servidor, hace su aparición Internet que puede ser observada como un soft al que pueden acceder usuarios individuales. Aparecen las tecnologías orientadas a objetos, los sistemas expertos y el software de inteligencia artificial.

Los problemas relacionados con el software han persistido a lo largo de la evolución de los sistemas informáticos y continúan aumentando.

1\_ Los avances del soft dejan atrás nuestra habilidad para construir software que pueda explotar el nuevo hardware.

2\_ Nuestra habilidad para programar no satisface la demanda de nuevos programas.

3\_ La sociedad se ha hecho cada vez dependiente de la operación fiable del software.

4\_ Luchamos por construir soft fiable y de alta calidad.

5\_ La habilidad de soportar y mejorar los programas existentes se ve amenazada por diseños pobres y recursos inadecuados.

# Una perspectiva industrial

Al principio el software se utilizaba para gestionar el hardware que era el factor principal en el presupuesto del proyecto. Por tal motivo, se aplicaron controles, métodos y herramientas conocidas como ingeniería del hardware y el software no era mas que un añadido.

En los primeros días, la programación se veía como un arte con pocos métodos formales. Hoy en día paso a ser el elemento principal del costo de un proyecto y los desarrolladores comenzaron a preguntarse:

1\_ ¿ Por qué lleva tanto tiempo crear programas ?

2\_ ¿ Por qué es tan elevado el costo ?

3\_ ¿ Porque no se pueden encontrar gran cantidad de errores antes de entregar el soft al cliente ?

Estas y otras tantas cuestiones han llevado a la adopción de la ingeniería del software.

# Una fabrica de software que envejece

La industria del soft se encuentra hoy en día en una situación similar a la de la industria del metal de los años cincuenta y sesenta donde hay cientos de aplicaciones de soft en una situación critica que necesitan ser renovadas urgentemente:

\_ Las aplicaciones escritas hace veinte años que son virtualmente imposibles de mantener e incluso una pequeña modificación puede hacer que falle todo el sistema.

\_ Las aplicaciones de ingeniería que se utilizan para generar datos críticos de diseño de las cuales nadie tiene un conocimiento detallado de su estructura interna de los programas.

\_ Sistemas empotrados (usados en fabricas, tráfico aéreo, etc.) que poseen un comportamiento extraño e inexplicable y no pueden ser puestos fuera de servicio ya que no poseen reemplazo.

Muchos de estos sistemas requieren de una reingeniería o reestructuración que muchos directores no parecen dispuestos a realizar.

# Competitividad del software

Durante muchos años los desarrolladores empleados por compañías eran los únicos del rubro. Hoy en día el hardware se ha convertido en un comodity y el software es una empresa competitiva con una gran competencia de profesionales altamente motivados, capacitados y de bajo costo.

# EL SOFTWARE

La definición estricta indica que el software es:

\_ Instrucciones que cuando se ejecutan proporcionan la función y el rendimiento deseados.

\_ Estructuras de datos que permiten a los programas manipular adecuadamente información y

\_ Documentos que describen la operación y uso de programas.

Pero nosotros necesitamos algo mas que una definición formal.

# Características del software

Cuando construimos hardware el proceso creativo humano (análisis, diseño, construcción, etc.) se traduce finalmente en una forma física. El software es un elemento del sistema que es lógico en lugar de físico por lo tanto:

**1\_** **Se desarrolla, no se fabrica**. La fase de construcción del hardware puede introducir problemas de calidad que no existen (o son fácilmente corregibles) en el software. Ambas actividades dependen de personas, pero la relación persona-trabajo es muy diferente. Ambas requieren la construcción de un producto, pero los métodos son diferentes.

Los costos del software se centran en la ingeniería y no se pueden gestionar como si fueran proyectos de fabricación.

**2\_ El software no se estropea**. En la figura se ve la proporción de fallos en función del tiempo para el hardware donde se percibe una gran cantidad al comienzo (defectos de diseño o fabricación), luego se estabilizan por un tiempo hasta que aparecen las fallas por desgaste, suciedad, vibración, etc.

En el soft los defectos no detectados hacen que el programa falle durante las primeras etapas de su vid, una vez que se corrigen (sin introducir nuevos errores), no vuelven a aparecer.

Pero si bien el software no se estropea, **se deteriora**. Se puede ver en la figura como los cambios por mantenimiento introducen nuevos defectos que antes de que desaparezcan se solicita un nuevo cambio que introduce nuevos errores y así sucesivamente teniendo el mantenimiento del soft una complejidad mayor al mantenimiento del hard.

**3\_ La mayoría del Soft se construye a medida en lugar de ensamblar componentes existentes**. Se puede comprar software ya desarrollado, pero sólo como una unidad completa, a menudo el ingeniero construye un circuito adquiriendo pieza como circuitos integrados que poseen una función definida, una interfaz también definida y criterios de integración que por desgracia no existen para el software.

**Componentes del Software**

Cuando se diseña nuevo hardware se utilizan CI’s, tornillos y otros componentes estándares reutilizables creados para que el ingeniero centre su trabajo en el diseño de lo verdaderamente innovador. La reutilización de componentes es algo que aun se tiene que lograr en una escala amplia en el mundo del soft.

Los componentes de soft se construyen mediante un lenguaje con un vocabulario limitado que en el nivel mas bajo se trata del lenguaje maquina (Assembler) que refleja un conjunto de instrucciones del hard. Los lenguajes de alto nivel permiten al programador y programa independizarse de la maquina.

# Aplicaciones del Software

El software puede aplicarse a cualquier situación en la que se haya previamente definido un conjunto especifico de pasos procedimentales (algoritmo) teniendo en cuenta el contenido (es decir el significado y forma de la información de entrada y salida) y el determinismo (predecibilidad del orden y del tiempo de llegada de los datos Ej: Aplicaciones determinadas (los datos llegan en orden y se ejecuta un proceso), Aplicaciones Indeterminadas (Multiusuario)) de la información.

# Software de Sistemas

Es el conjunto de programas diseñado para servir a otros programas.

Compiladores, Editores, Gestión de Archivos, etc. son ejemplos de aplicaciones determinadas. Componentes del SO, manejo de periféricos, procesadores de comunicaciones, etc. son ejemplos de aplicaciones indeterminadas. Este área se caracteriza por la fuerte interacción con el hard, gran utilización por múltiples usuarios, comparición de recursos, estructuras de datos complejas y múltiples interfaces de salida.

## Software de tiempo real

Mide, analiza y controla sucesos del mundo real conforme ocurren. Cuenta con un componente de adquisición de datos que recolecta la información del entorno, un componente de análisis que transforma la información según lo requiera la aplicación, un componente de control/ salida que responda al entorno externo y un componente de monitorización que coordina a todos los demás componente, de forma que pueda mantenerse la respuesta en tiempo real.

# Software de gestión

El procesamiento de info comercial constituye la mayor de las áreas de aplicación del software. Los sistemas discretos (nominas, inventarios, etc.) han evolucionado hacia el soft de sistemas de información de gestión que accede a una o más bases de datos que contiene la información comercial y la reestructuran para facilitar la toma de decisiones.

# Software de ingeniería y científico

Esta caracterizado por los algoritmos de manejo de números. Las aplicaciones van desde astronomía, automotores, biología molecular hasta soft con características de tiempo real en CAD’s , simuladores y otras aplicaciones interactivas.

# Software Empotrado

Reside en la memoria de solo lectura y se utiliza para controlar productos y sistemas de los mercados industriales y de consumo (funciones de un horno a microondas, control de consumo de nafta, etc.).

# Soft de PC’s

Procesadores de texto, planillas de calculo, multimedia, juegos, educativos, gestión de BD’s.

# Soft de inteligencia artificial

Hace uso de algoritmos no numéricos para resolver problemas complejos para los que no son adecuados él calculo o análisis directo. El área de mayor difusión son los sistemas expertos o basados en conocimiento pero también se utiliza en reconocimiento de patrones (voz, imágenes, etc.), juegos, etc. Y últimamente en las redes neuronales artificiales que simulan la estructura de proceso del cerebro.

# SOFTWARE: UNA CRISIS EN EL HORIZONTE

La definición de crisis como el punto decisivo en el curso de una enfermedad, cuando se ve mas claro si un paciente vivirá o morirá puede darnos una pista sobre los problemas que han acosado al desarrollo del software.

Dado que los problemas no se limitan al soft que no funciona, es mas, abarca los problemas asociados a como desarrollar software, como mantener el volumen de soft existente y como afrontar la demanda creciente podemos hablar de una aflicción crónica mas que de una crisis.

# MITOS DEL SOFTWARE

Muchas de las causas de la crisis del software pueden atribuirse a viejos mitos que aun hoy persisten.

# Mitos de gestión

Los gestores poseen la presión de cumplir con tiempos, presupuestos, etc. con lo cual los gestores se aferran a mitos para disminuir las presiones temporalmente.

Mito\_ Tenemos un manual lleno de estándares y procedimientos para construir software / No le proporciona mi gente todo lo que necesita saber ?

Realidad\_ Los libros generalmente existen, pero se usan, están actualizados, son completos, etc.

Mito\_ Mi gente dispone de las herramientas más avanzadas ya que posee las PC’s más modernas.

Realidad\_ Se necesita mucho mas que las mejores PC’s para el desarrollo de soft. Las herramientas CASE de ingeniería de soft asistido por PC suelen ser más importantes que el hard.

Mito\_ Si fallamos en la planificación, podemos añadir mas programadores y recuperar el tiempo perdido.

Realidad\_ Añadir gente a un proyecto de soft atrasado lo retrasa mas. La necesidad de aprender y comunicarse con el resto del equipo afecta al tiempo de desarrollo productivo.

Mitos del Cliente

Mito\_ Una declaración general de los objetivos es suficiente para comenzar a programar

Realidad\_ Es esencial una descripción formal y detallada del ámbito de la información, funciones, rendimiento, interfaces, etc. Que pueden determinarse solo después de una exhaustiva comunicación entre el cliente y el analista.

Mito\_ Los requisitos del proyecto cambian continuamente, pero los cambios pueden acomodarse fácilmente, ya que el soft es flexible.

Realidad\_ El impacto del cambio varia según el momento en que se introduzcan. Si se pone cuidado al dar la definición inicial, los cambios solicitados al principio pueden acomodarse fácilmente. Los cambios sobre la etapa de diseño hacen aumentar el impacto rápidamente dado que ya se han acordado los recursos y el esqueleto del diseño. Los cambios sobre la implementación tienen un orden de magnitud mas caro que el mismo cambio solicitado al principio.

Mitos de los desarrolladores

Mito\_ Una vez que escribimos el programa y hacemos que funcione, nuestro trabajo ha terminado.

Realidad\_ Cuanto más rápido se comience a escribir el código, mas tarde se termina. Entre el 50 y 60% de todo el esfuerzo dedicado a un programa se realizará después de entregado al cliente por primera vez.

Mito\_ Hasta que no tengo el programa ejecutándose, realmente no tengo forma de comprobar su calidad.

Realidad\_ Desde el principio del proyecto se puede aplicar la revisión técnica formal que es un filtro de calidad mas que efectivo.

Mito\_ Lo único que se entrega al terminar el proyecto es el programa funcionando

Realidad\_ Un programa que funciona es solo una parte de una configuración del software que contiene programas, documentos y datos. La documentación es la base de un buen desarrollo y proporciona guías para la tarea de mantenimiento.