

INGENIERÍA DE SOFTWARE Y REQUERIMIENTOS

FACULTAD DE INGENIERÍA EN SISTEMAS

LA NATURALEZA DEL SOFTWARE

LA NATURALEZA DEL SOFTWARE

- En la actualidad, el software tiene un papel dual.
 - Es un producto y al mismo tiempo es el vehículo para entregar un producto

LA NATURALEZA DEL SOFTWARE

- En su forma de producto, brinda el potencial de cómputo incorporado en el hardware de cómputo o, con más amplitud, en una red de computadoras a las que se accede por medio de un hardware local.
- Ya sea que resida en un teléfono móvil u opere en el interior de una computadora central, el software es un transformador de información
 - Produce, administra, adquiere, modifica, despliega o transmite información que puede ser tan simple como un solo bit o tan compleja como una presentación con multimedios generada a partir de datos obtenidos de decenas de fuentes independientes

LA NATURALEZA DEL SOFTWARE

- Como vehículo utilizado para distribuir el producto, el software actúa como la base para el control de la computadora (sistemas operativos), para la comunicación de información (redes) y para la creación y control de otros programas (herramientas y ambientes de software).

LA NATURALEZA DEL SOFTWARE

- El software distribuye el producto más importante de nuestro tiempo:
 - *Información.*
- Transforma los datos personales (por ejemplo, las transacciones financieras de un individuo) de modo que puedan ser más útiles en un contexto local.
- Administra la información de negocios para mejorar la competitividad
- Provee una vía para las redes mundiales de información (la internet) y brinda los medios para obtener información en todas sus formas.

LA NATURALEZA DEL SOFTWARE

- En el último medio siglo, el papel del software de cómputo ha sufrido un cambio significativo
- Actualmente existen sistemas basados en computadora más sofisticados y complejos gracias a:
 - Las notables mejoras en el funcionamiento del hardware
 - Los profundos cambios en las arquitecturas de computadora,
 - El gran incremento en la memoria y capacidad de almacenamiento, y
 - Una amplia variedad de opciones de entradas y salidas exóticas
- Cuando un sistema tiene éxito la sofisticación y complejidad producen resultados deslumbrantes, pero también plantean problemas enormes para aquellos que deben construir sistemas complejos

LA NATURALEZA DEL SOFTWARE

- En la actualidad, la enorme industria del software se ha convertido en un factor dominante en las economías del mundo industrializado
- Equipos de especialistas de software, cada uno centrado en una parte de la tecnología que se requiere para llegar a una aplicación compleja, han reemplazado al programador solitario de los primeros tiempos
- A pesar de ello, las preguntas que se hacía aquel programador son las mismas que surgen cuando se construyen sistemas modernos basados en computadora

LA NATURALEZA DEL SOFTWARE



Preguntas que se hacen programadores cuando se construyen sistemas modernos basados en computadora:

- ¿Por qué se requiere tanto tiempo para terminar el software?
- ¿Por qué son tan altos los costos de desarrollo?
- ¿Por qué no podemos detectar todos los errores antes de entregar el software a nuestros clientes?
- ¿Por qué dedicamos tanto tiempo y esfuerzo a mantener los programas existentes?
- ¿Por qué seguimos con dificultades para medir el avance mientras se desarrolla y mantiene el software?

Éstas y muchas otras preguntas, denotan la preocupación sobre el software y la manera en que se desarrolla, preocupación que ha llevado a la **adopción de la práctica de la ingeniería del software**.

LA NATURALEZA DEL SOFTWARE



- Certificación en Requerimientos:
 - [Home - IREB – International Requirements Engineering Board](#)
- Gestión de Proyectos
 - [PMBOK Guide | Project Management Institute \(pmi.org\)](#)

142.000 M€

de pérdidas anuales en Europa
por proyectos fallidos

66%

van por encima de
presupuesto

1 de 6

tiene un sobrecoste del
200% y un retraso del
70%

**“Hasta un 70% de los
proyectos de software no salen
bien”**

33%

van por encima de
tiempo

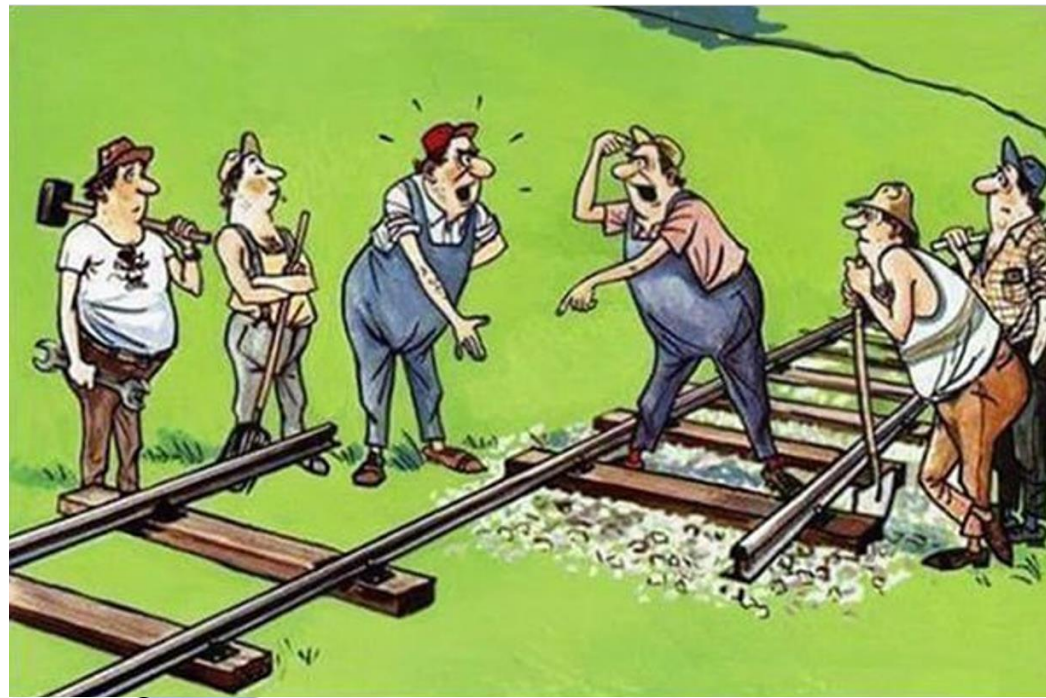
75%

de ejecutivos no confían
en que salga bien

17%

van tan mal que ponen en riesgo la
existencia de la empresa

www.techbizdesign.com



CONCEPTOS BÁSICOS DE SOFTWARE

Ingeniería	Software Ing. SW	Principios Fundamentos Patrones
Metodología	Método	Proceso
Framework	Técnicas	Procedimiento
Productos	Fases	Actividades Sistemas
Proyectos de SW		Aplicaciones
Herramientas		Componentes

DEFINICIÓN DE SOFTWARE

El software es:

- 1) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados;
- 2) estructuras de datos que permiten que los programas manipulen en forma adecuada la información, y
- 3) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas.

DEFINICIÓN DE SOFTWARE

El software es:

- Programas de computadora, procedimientos y posiblemente documentación y datos asociados, referentes a la operación de un sistema de computación.

DEFINICIÓN DE SOFTWARE

¿Cuáles son los atributos del buen software?

- El buen software debe entregar al usuario la funcionalidad y el desempeño requeridos, y debe ser sustentable, confiable y utilizable.

DEFINICIÓN DE SOFTWARE

¿Qué es Ingeniería?

- Conjunto de conocimientos técnicos y científicos orientados a la invención y utilización de técnicas para el aprovechamiento de los recursos naturales o para la actividad industrial.
- Es la disciplina que aplica los conocimientos científicos y tecnológicos para diseñar, construir, mantener y mejorar estructuras, máquinas, sistemas y procesos que satisfacen las necesidades de la sociedad.

DEFINICIÓN DE SOFTWARE

¿Qué es un ingeniero?

- Persona legalmente capacitada para ejercer la ingeniería. Los ingenieros son como los arquitectos del mundo moderno. Utilizan su creatividad y conocimientos técnicos para encontrar soluciones a problemas reales y mejorar nuestra calidad de vida.

DEFINICIÓN DE SOFTWARE

¿Qué hace un ingeniero?

- **Diseña:** Crea planos y modelos de todo tipo de cosas, desde edificios hasta software.
- **Construye:** Supervisa y gestiona la construcción de proyectos.
- **Analiza:** Utiliza herramientas matemáticas y computacionales para resolver problemas.
- **Innova:** Desarrolla nuevas tecnologías y mejora las existentes.
- **Resuelve problemas:** Encuentra soluciones eficientes y prácticas a desafíos complejos.

DEFINICIÓN DE SOFTWARE

¿Qué es Ingeniería de Software?

- La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software.

DEFINICIÓN DE SOFTWARE

¿Según Ian Somerville, cuáles son **las actividades fundamentales** de la ingeniería de software?

- Especificación,
- desarrollo,
- validación y
- evolución del software.

DEFINICIÓN DE SOFTWARE

¿Cuál es la diferencia entre ingeniería de software y ciencias de la computación?

- Las ciencias de la computación se enfocan en teoría y fundamentos;
- mientras la ingeniería de software se enfoca en el sentido práctico del desarrollo y en la distribución de software.

DEFINICIÓN DE SOFTWARE

¿Cuál es la diferencia entre ingeniería de software e ingeniería de sistemas?

- La ingeniería de sistemas se interesa por todos los aspectos del desarrollo de sistemas basados en computadoras, incluidos hardware, software e ingeniería de procesos.
- La ingeniería de software es parte de este proceso más general.

DEFINICIÓN DE SOFTWARE

¿Cuáles son los principales retos que enfrenta la ingeniería de software?

- Se enfrentan con una **diversidad creciente**, demandas por tiempos de distribución limitados y desarrollo de software confiable.

- [Stack Overflow Developer Survey 2023](#)
- [State of JavaScript 2022: Front-end Frameworks \(stateofjs.com\)](#)
- [React, Angular - Explorar - Tendencias de Google](#)

DEFINICIÓN DE SOFTWARE

¿Cuáles son los costos de la ingeniería de software?

- Aproximadamente 60% de los costos del software son de desarrollo, y 40% de prueba.
- Para el software elaborado específicamente, los costos de evolución superan con frecuencia los costos de desarrollo.

DEFINICIÓN DE SOFTWARE

¿Cuáles son los mejores métodos y técnicas de la ingeniería de software?

- Aun cuando todos los proyectos de software deben gestionarse y desarrollarse de manera profesional, existen diferentes técnicas que son adecuadas para distintos tipos de sistema.
 - Por ejemplo, los juegos siempre deben diseñarse usando una serie de prototipos,
 - mientras que los sistemas críticos de control de seguridad requieren de una especificación completa y analizable para su desarrollo.
 - Por lo tanto, no puede decirse que un método sea mejor que otro.

DEFINICIÓN DE SOFTWARE

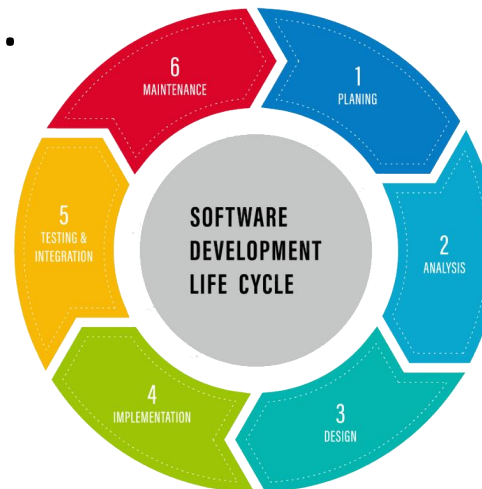
¿Qué diferencias ha marcado la Web a la ingeniería de software?

- La Web ha llevado a la disponibilidad de servicios de software y a la posibilidad de desarrollar sistemas basados en servicios distribuidos ampliamente.
- El desarrollo de sistemas basados en Web ha conducido a importantes avances en lenguajes de programación y reutilización de software.

DEFINICIÓN DE SOFTWARE

Ciclo de vida de desarrollo de software:

- Periodo de tiempo que comienza con la decisión de desarrollar un producto software y termina cuando el software es entregado.
- Típicamente incluye las fases de requerimientos, diseño, implementación y pruebas, y en ocasiones también las fases de instalación y puesta en marcha.



DEFINICIÓN DE SOFTWARE

Proceso de software es una secuencia de actividades que conducen a la elaboración de un producto de software.

- Cada actividad posee unas entradas y unas salidas.
- Las salidas son productos parciales que forman parte del producto final el sistema software funcionando y entregado al cliente.

DEFINICIÓN DE SOFTWARE

Producto de software:

- Conjunto completo de programas de computadora, procedimientos, y posiblemente documentación y datos asociados, diseñados para ser entregados al usuario

DEFINICIÓN DE SOFTWARE

Aplicación:

- Es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas.
- Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas.

DEFINICIÓN DE SOFTWARE

Sistema:

- Conjunto de componentes o elementos (aplicaciones) con características y funcionalidad propias, que están relacionados y que trabajan en conjunto con el fin de alcanzar un objetivo, resultado o resolver un problema.

DEFINICIÓN DE SOFTWARE

Un sistema de software incluye:

1. Componentes funcionales o funciones
2. Estructuras de datos o configuración de datos
3. Documentación
4. Interfaces
5. Componentes no funcionales
6. Componentes dinámicos

DEFINICIÓN DE SOFTWARE

Es importante también añadir aquellas características del software que lo hacen diferente de otros objetos que construyen los seres humanos, ya que el software no tiene características físicas como el hardware:

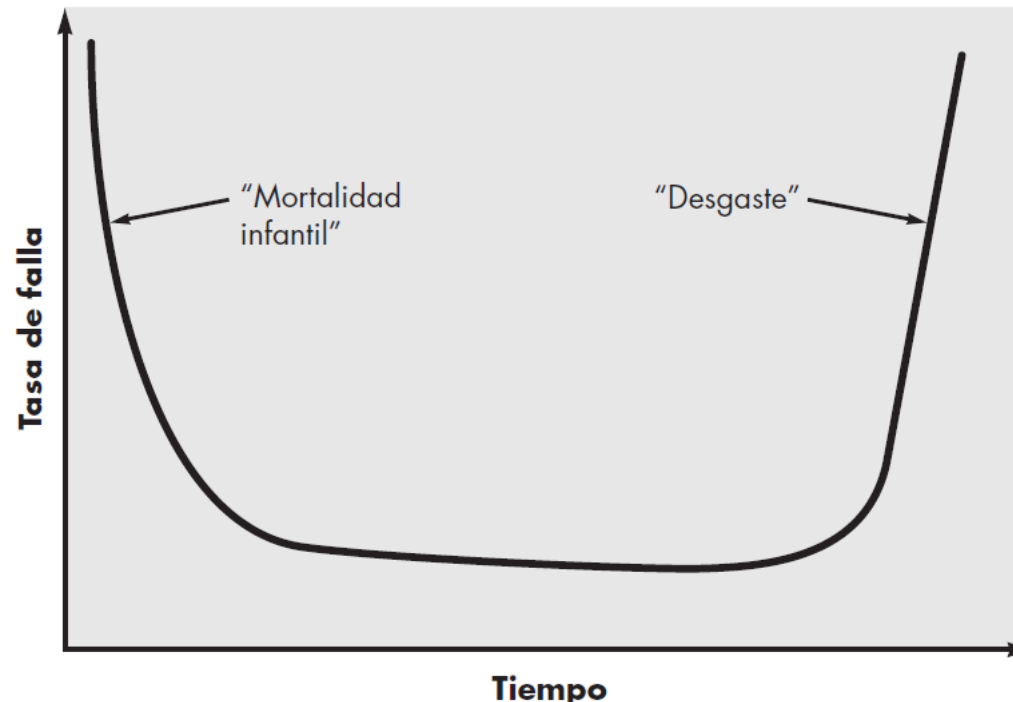
1. El software se desarrolla o modifica con intelecto, no se manufactura en el sentido clásico.
2. Aunque la industria se mueve hacia la construcción basada en componentes, la mayor parte del software se construye para un uso individualizado

DEFINICIÓN DE SOFTWARE

El software no se “desgasta”

- En la siguiente se ilustra la curva de la tasa de fallas del hardware como función del tiempo.

defectos de diseño
o manufactura

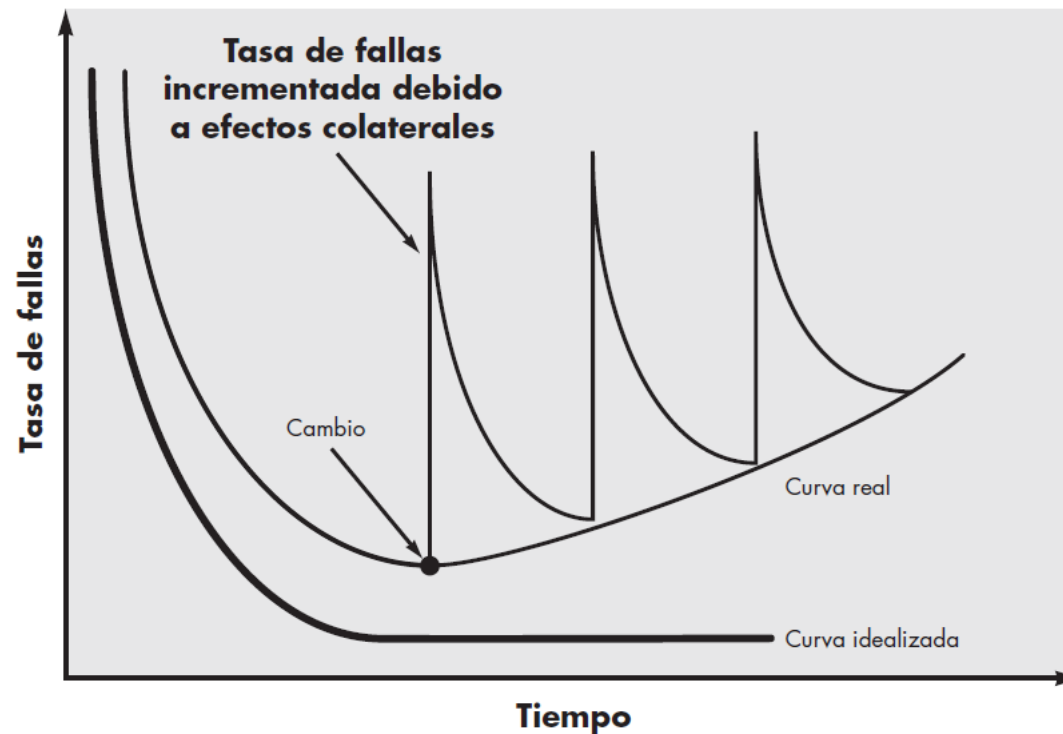


suciedad, vibración,
abuso, temperaturas extremas

DEFINICIÓN DE SOFTWARE

El software no se desgasta, ¡pero sí se deteriora!

- En la siguiente figura se ilustra la curva de la tasa de fallas del software en función del tiempo



Curvas de falla del software

DEFINICIONES DE INGENIERÍA DE SOFTWARE

INGENIERÍA DE SOFTWARE

1. Es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.
2. Ingeniería de Software es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas software. (Zelkovitz 1978)
 - Estudio de fundamentos

INGENIERÍA DE SOFTWARE

3. Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y la construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantener esos programas. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).
4. Ingeniería de Software trata del establecimiento y uso de los principios y métodos de la ingeniería, a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales (Bauer, 1972).

INGENIERÍA DE SOFTWARE

5. La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software, es decir, la aplicación de la ingeniería al software (IEEE, 2016).

INGENIERÍA DE SOFTWARE

- Tradicionalmente, la Ingeniería de Software ha sido una expresión aplicada para la **gestión de proyectos de desarrollo de software**, involucrando el uso de un número de personas distintas en papeles diferentes.
- Junto con la Ingeniería de Software, han sido aplicadas técnicas gerenciales, metodológicas y métricas para producir una gran cantidad de productos confiables, mantenibles, reutilizables, dentro de presupuesto y a tiempo (RINE, 1993).

INGENIERÍA DE SOFTWARE

Thayer (1990):

- (1) La aplicación práctica de las ciencias de la computación, de la administración y de otras ciencias para el análisis, diseño, construcción y mantenimiento de software y su documentación asociada.
- (2) Una ciencia de ingeniería que aplica los conceptos de análisis, diseño, codificación, pruebas, documentación y gerenciamiento para obtener con éxito grandes programas de computador contruidos para los usuarios.
- (3) La aplicación sistemática de métodos, herramientas y técnicas para cumplir con el requisito u objetivo establecido para un efectivo y eficiente sistema de software.

DOMINIOS DE APLICACIÓN DEL SOFTWARE

DOMINIOS DE APLICACIÓN DEL SOFTWARE

Actualmente hay siete grandes categorías de software de computadora:

1. Software de sistemas
2. Software de aplicación
3. Software de ingeniería y ciencias
4. Software incrustado
5. Software de línea de productos
6. Aplicaciones web
7. Software de inteligencia artificial

1. SOFTWARE DE SISTEMAS

- Conjunto de programas escritos para dar servicio a otros programas.
- Determinado software de sistemas que procesa estructuras de información complejas pero deterministas.
 - Por ejemplo, compiladores, editores y herramientas para administrar archivos
- Otras aplicaciones de sistemas que procesan sobre todo datos indeterminados.
 - Por ejemplo, componentes de sistemas operativos, software de redes, procesadores de telecomunicaciones.

1. SOFTWARE DE SISTEMAS

Características:

- Gran interacción con el hardware de la computadora
- Uso intensivo por parte de usuarios múltiples
- Operación concurrente que requiere la secuenciación
- Recursos compartidos y administración de un proceso sofisticado
- Estructuras complejas de datos e interfaces externas múltiples.

2. SOFTWARE DE APLICACIÓN

- Programas aislados que resuelven una necesidad específica de negocios.
- Las aplicaciones en esta área procesan datos comerciales o técnicos en una forma que facilita las operaciones de negocios o la toma de decisiones administrativas o técnicas.
- Además de las aplicaciones convencionales de procesamiento de datos, el software de aplicación se usa para controlar funciones de negocios en tiempo real
 - Por ejemplo, procesamiento de transacciones en punto de venta, control de procesos de manufactura en tiempo real.

3. SOFTWARE DE INGENIERÍA Y CIENCIAS

- Se ha caracterizado por usar algoritmos “devoradores de números”.
- Las aplicaciones van de la astronomía a la vulcanología, del análisis de tensiones en automóviles a la dinámica orbital del transbordador espacial, y de la biología molecular a la manufactura automatizada.
- Sin embargo, las aplicaciones modernas dentro del área de la ingeniería y las ciencias están abandonando los algoritmos numéricos convencionales.
- El diseño asistido por computadora, la simulación de sistemas y otras aplicaciones interactivas, han comenzado a hacerse en tiempo real e incluso han tomado características del software de sistemas.

4. SOFTWARE INCRUSTADO O EMBEBIDO

- Reside dentro de un producto o sistema y se usa para implementar y controlar características y funciones para el usuario final y para el sistema en sí.
- El software incrustado ejecuta funciones limitadas y particulares (por ejemplo, control del tablero de un horno de microondas) o provee una capacidad significativa de funcionamiento y control (funciones digitales en un automóvil, como el control del combustible, del tablero de control y de los sistemas de frenado).

5. SOFTWARE DE LÍNEA DE PRODUCTOS

- Es diseñado para proporcionar una capacidad específica para uso de muchos consumidores diferentes.
- El software de línea de productos se centra en algún mercado limitado y particular (por ejemplo, control del inventario de productos) o se dirige a mercados masivos de consumidores (procesamiento de textos, hojas de cálculo, gráficas por computadora, multimedios, entretenimiento, administración de base de datos y aplicaciones para finanzas personales o de negocios).

6. APLICACIONES WEB

- Llamadas “webapps”, esta categoría de software centrado en redes agrupa una amplia gama de aplicaciones.
- En su forma más sencilla, las webapps son poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas.
- Sin embargo, desde que surgió Web 2.0, las webapps están evolucionando hacia ambientes de cómputo sofisticados que no sólo proveen características aisladas, funciones de cómputo y contenido para el usuario final, sino que también están integradas con bases de datos corporativas y aplicaciones de negocios.

7. SOFTWARE DE INTELIGENCIA ARTIFICIAL

- Hace uso de algoritmos no numéricos para resolver problemas complejos que no son fáciles de tratar computacionalmente o con el análisis directo.
- Las aplicaciones en esta área incluyen
 - robótica,
 - sistemas expertos,
 - reconocimiento de patrones (imagen y voz),
 - redes neurales artificiales,
 - demostración de teoremas y juegos.

NUEVAS CATEGORÍAS

Millones de ingenieros de software en todo el mundo los que trabajan duro en proyectos de software en una o más de estas categorías. Nuevos desafíos han aparecido en el horizonte.

- **Computación en un mundo abierto:** el rápido crecimiento de las redes inalámbricas quizá lleve pronto a la computación verdaderamente ubicua y distribuida.
- **Construcción de redes:** la red mundial (World Wide Web) se está convirtiendo con rapidez tanto en un motor de computación como en un proveedor de contenido.
- **Fuente abierta (Open Source):** tendencia creciente que da como resultado la distribución de código fuente para aplicaciones de sistemas (por ejemplo, sistemas operativos, bases de datos y ambientes de desarrollo) de modo que mucha gente pueda contribuir a su desarrollo.

SOFTWARE HEREDADO/LEGADO (LEGACY SOFTWARE)

- Software desarrollado hace muchas décadas y que continúa siendo utilizado porque continúan siendo un apoyo para las funciones básicas del negocio y son “indispensables” para éste.
- El software heredado se caracteriza por su longevidad e importancia crítica para el negocio, pero en muchas ocasiones son de mala calidad.

SOFTWARE HEREDADO/LEGADO (LEGACY SOFTWARE)

- Software desarrollado hace muchas décadas y que continúa siendo utilizado porque continúan siendo un apoyo para las funciones básicas del negocio y son “indispensables” para éste.
- El software heredado se caracteriza por su longevidad e importancia crítica para el negocio
- No son susceptibles de extenderse, código confuso, documentación mala o inexistente, casos y resultados de pruebas que nunca se archivaron, una historia de los cambios mal administrada.
- Si el software heredado satisface las necesidades de sus usuarios y corre de manera confiable, entonces no falla ni necesita repararse

SOFTWARE HEREDADO/LEGADO (LEGACY SOFTWARE)

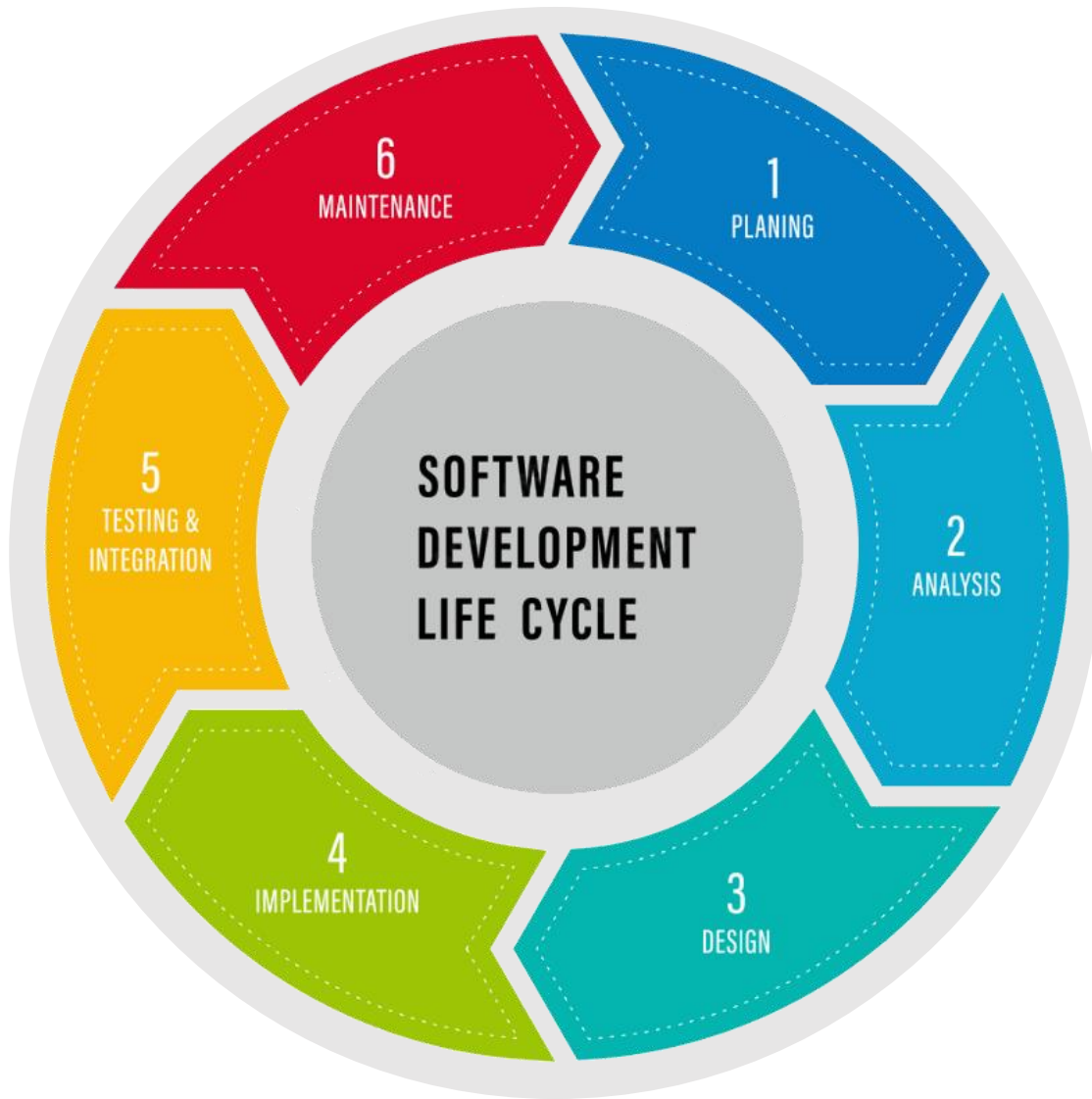
Conforme pasa el tiempo será frecuente que los sistemas de software evolucionen por una o varias de las siguientes razones:

- El software debe adaptarse para que cumpla las necesidades de los nuevos ambientes del cómputo y de la tecnología.
- El software debe ser mejorado para implementar nuevos requerimientos del negocio.
- El software debe ampliarse para que sea operable con otros sistemas o bases de datos modernos.
- La arquitectura del software debe rediseñarse para hacerla viable dentro de un ambiente de redes.

SOFTWARE HEREDADO/LEGADO (LEGACY SOFTWARE)

Cuando ocurren estos modos de evolución, debe hacerse la reingeniería del sistema heredado (legado) para que sea viable en el futuro.

La meta de la ingeniería de software moderna es “**desarrollar metodologías que se basen en el concepto de evolución**”; es decir, el concepto de que los sistemas de software cambian continuamente, que los nuevos sistemas de software se desarrollan a partir de los antiguos y que todo debe operar entre sí y cooperar con cada uno de los demás” [Day99].



THANK YOU



VICTOR.VELEPUCHA@EPN.EDU.EC

