UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

Centro Universitario UAEM Nezahualcóyotl

Licenciatura en Ingeniería en Sistemas Inteligentes

UNIDAD DE APRENDIZAJE:
INGENIERÍA DEL SOFTWARE

Unidad I:
Introducción a la Ingeniería del Software

Dra. Carmen Liliana Rodríguez Páez
clrodriguezp@uaemex.mx

CU NEZAHUALCÓYOTL

Directorio Institucional

	Rector
Doctor en Educación	**Alfredo Barrera Baca**
Maestra en Salud Pública	
María Estela Delgado Maya	Secretaria de Docencia
Doctor en Ciencias e Ingeniería	
Carlos Eduardo Barrera Díaz	Secretario de Investigación y Estudios Avanzados
Doctor en Ciencias Sociales	
Luis Raúl Ortiz Ramírez	Secretario de Rectoría
Doctor en Arte	
José Édgar Miranda Ortiz	Secretario de Difusión Cultural
Maestra en Comunicación	
Jannet Valero Vilchis	Secretaria de Extensión y Vinculación
Maestro en Economía	
Javier González Martínez	Secretario de Administración
Doctor en Ciencias de la Computación	
José Raymundo Marcial Romero	Secretario de Planeación y Desarrollo Institucional
Maestra en Lingüística Aplicada	
María del Pilar Ampudia García	Secretaria de Cooperación Internacional
Doctora en Ciencias Sociales y Políticas	
Gabriela Fuentes Reyes	Abogada General
Licenciado en Comunicación	
Gastón Pedraza Muñoz	Director General de Comunicación Universitaria
Maestro en Relaciones Interinstitucionales	
Jorge Bernáldez García	Secretario Técnico de la Rectoría
Maestra en Administración Publica	
Guadalupe Ofelia Santamaría González	Directora General de Centros Universitarios y Unidades profesionales
Maestro en Administración	
Ignacio Gutiérrez Padilla	Contralor

CUNEZAHUALCÓYOTL

Directorio Centro Universitario UAEM Nezahualcóyotl

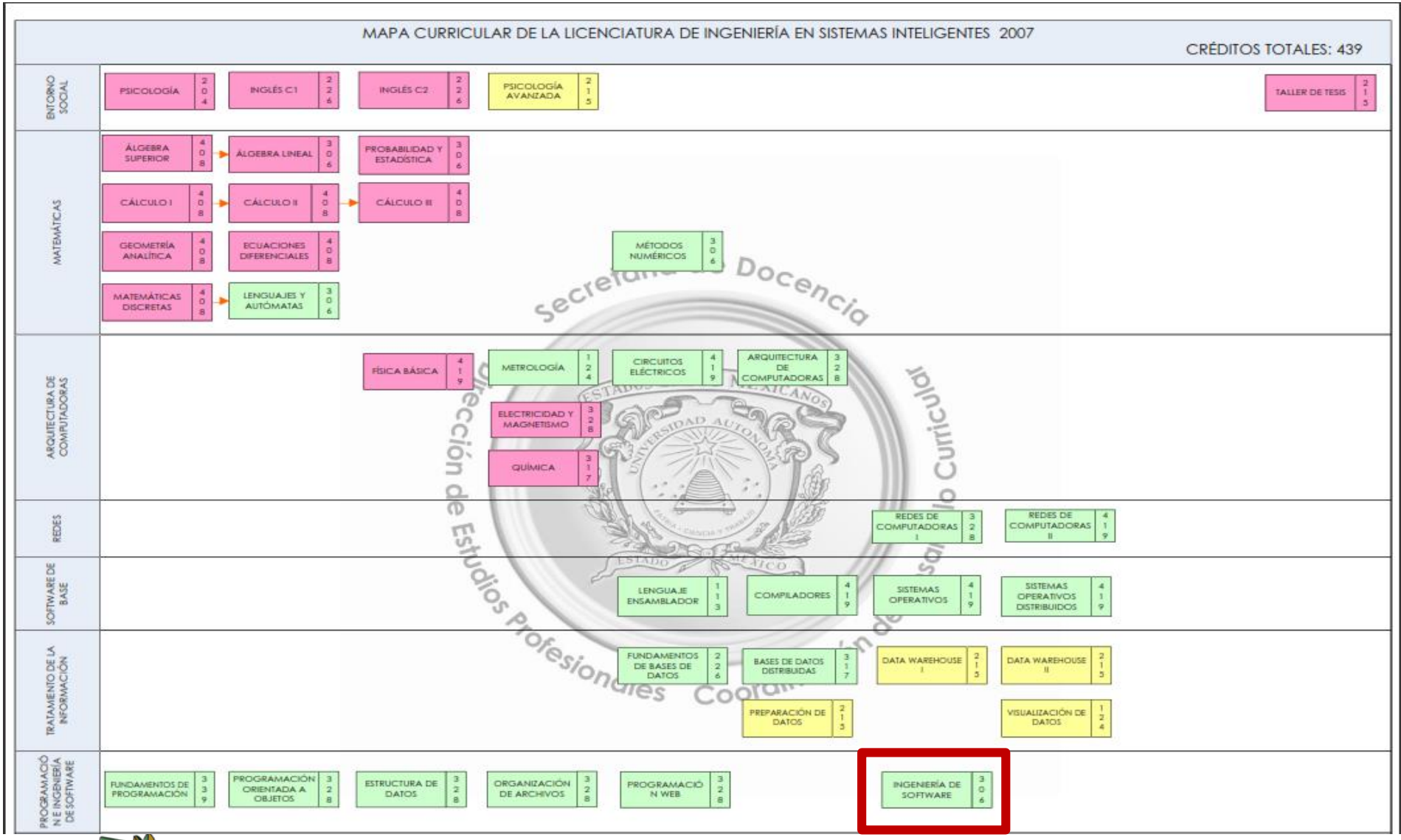
Maestro en Derecho **Juan Carlos Medina Huicochea**
Encargado del Despacho de la Dirección del Centro Universitario UAEM Nezahualcóyotl

Maestro en Ciencias **José Antonio Castillo Jiménez**	Subdirector Académico
Licenciado en Economía **Ramón Vital Hernández**	Subdirector Administrativo
Doctora en Ciencias Sociales **María Luisa Quintero Soto**	Coordinadora de Investigación y Estudios Avanzados
Licenciado en Administración de Empresas **Víctor Manuel Durán López**	Coordinador de Planeación y Desarrollo Institucional
Doctor en Relaciones Internacionales **Rafael Alberto Durán Gómez**	Coordinador de la Licenciatura en Comercio Internacional
Maestra en Ciencias **Gabriela Kramer Bustos**	Coordinadora de la Licenciatura en Educación para la Salud
Doctor en Ingeniería de los Sistemas **Ricardo Rico Molina**	Coordinador de la Licenciatura en Ingeniería en Sistemas Inteligentes
Doctor en Urbanismo **Noé Gaspar Sánchez**	Coordinador de la Licenciatura en Ingeniería en Transporte
Maestro en Ciencias de la Computación **Erick Nicólas Cabrera Álvarez**	Coordinador de la Licenciatura en Seguridad Ciudadana

CU NEZAHUALCÓYOTL

Ubicación de la asignatura Ingeniería del Software dentro del programa de la Lic. en Ingeniería en Sistemas Inteligentes

Mapa curricular



CUNEZAHUALCÓYOTL

Unidad de Aprendizaje

La Unidad de Aprendizaje (UA) de ***Ingeniería del Software*** tiene como área curricular la programación e ingeniería de software y forma parte del núcleo sustantivo.

Objetivo de la UA

Proporcionar los conocimientos y el desarrollo de habilidades que le permitirán proponer y analizar los elementos básicos del desarrollo de un producto de software concentrándose en las fases, técnicas y herramientas, que intervienen en la construcción de un Sistema de Información Basado en Computadoras (SIBC).

Presentación

El presente Material tiene como objetivo cubrir la primera unidad Introducción a la ingeniería del software del programa por competencia.

El alumno, será capa de realizar el sistema de información utilizando el ciclo de vida clásico de un software basado en la Metodología SIBC, la cual consta de cuatro fases: Análisis, Diseño, Desarrollo y Mantenimiento del Sistemas de Información.

Estructura de la Unidad de Aprendizaje

Unidad I. INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

Objetivo: Identificar los conceptos básicos de la ingeniería del software

Contenidos:

1.0 Introducción
1.1. Conceptos básicos.
1.2. La importancia de la ingeniería del software.
1.3. Historia de la Ingeniería del Software.
1.4 Descripción del ciclo de desarrollo de un producto de software
1.4.1 Etapas de desarrollo

Unidad II. ANÁLISIS DE REQUERIMIENTOS DEL SOFTWARE

Objetivo: Conocer e identificar los requerimientos del medio ambiente

Contenidos:

3.1 Concepto y características

3.2 Conocimiento del medio ambiente o investigación preliminar: identificar y obtener la visión, misión, planes, estrategias, objetivos, funciones y actividades de la empresa, estructura organizacional, identificar el área en particular donde se aplicará el sistema de información, identificar las funciones del área de oportunidad, realizar el diagrama de flujo de datos del funcionamiento del sistema actual, identificación de los procesos actuales.

3.3 identificación y análisis de necesidades: identificar y analizar los elementos actuales usando una tabla sistémica.

3.4 propuesta general de solución: elaborar diagrama de gantt y pert, elaborar marco normativo del sistema futuro, nuevo diagrama de flujo de datos, realizar el análisis costo beneficio del nuevo sistema, definir elementos del nuevo sistema.

Unidad II. MODELOS DE LA INGENIERÍA DE SOFTWARE

Objetivo: Conocer los modelos de la ingeniería del software.

Contenidos:

2.1. Modelo de capacidad de madurez.
2.2. Marco de trabajo para el proceso.
2.3. Modelos de la ingeniería del software: modelo de cascada, modelo de prototipos, modelo de espiral, modelo de Proceso Unificado Racional (RUP).
2.4. Tendencias modernas de modelos de la ingeniería del software.

Unidad IV. DISEÑO DE SOFTWARE

Objetivo: Realizar el diseño de una problemática del software

Contenidos:

4.Definición de los objetivos en esta etapa de desarrollo
4.2Definición de las actividades en la etapa de diseño
4.3Técnicas para el diseño de salidas
4.4Técnicas para la elaboración de modelos de datos.
4.5Técnicas para el diseño de Entradas
4.6Técnicas ara el Diseño de Procedimientos

Unidad V. PROGRAMACIÓN Y PRUEBAS DEL SOFTWARE

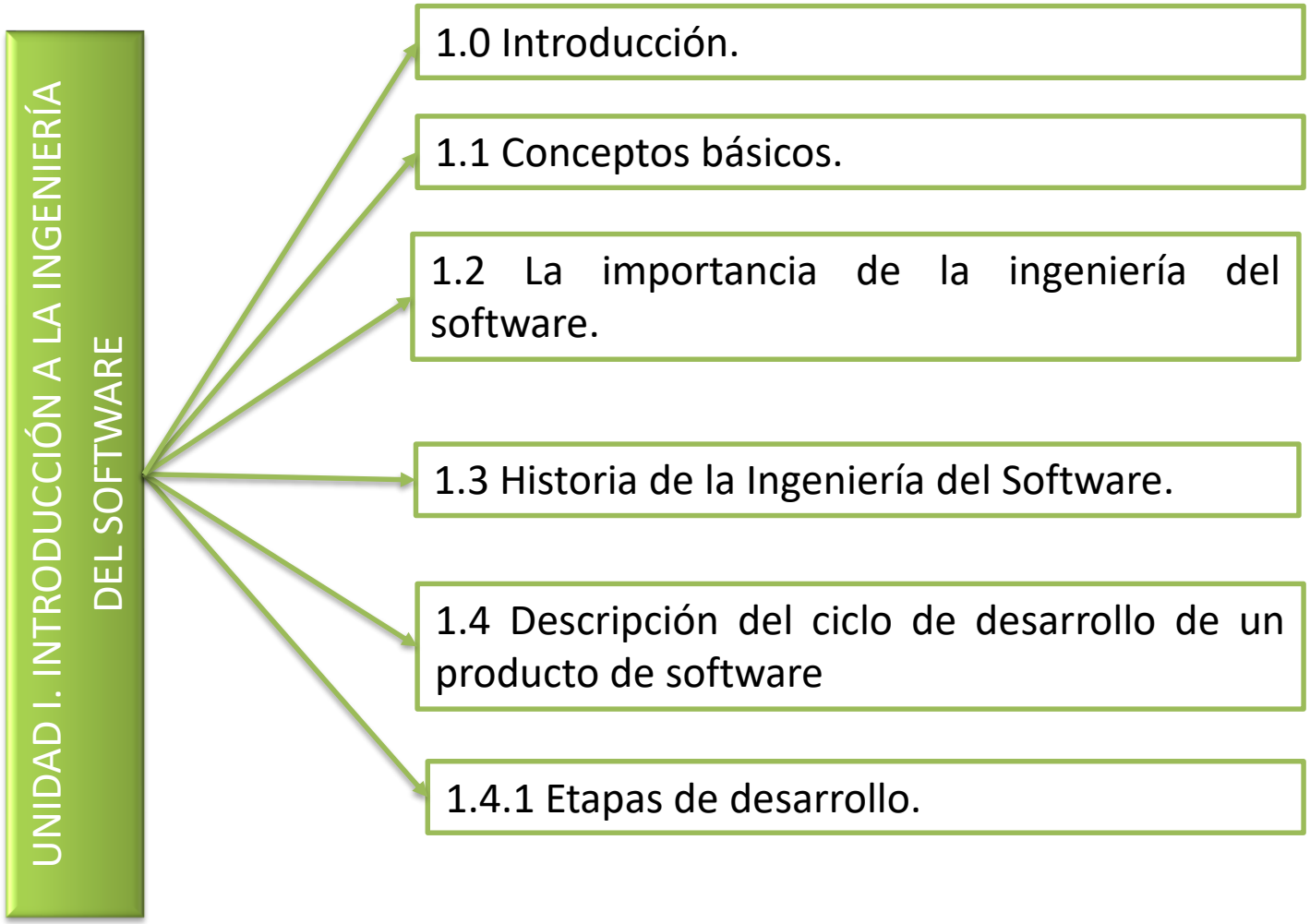
Objetivo: Desarrollo e implementación de las pruebas del software

Contenidos

5.1 Principios de construcción.
5.2 Codificación
5.3 Reutilización e Integración de código
5.4 Fallos, errores y defectos
5.5 Niveles y tipos de pruebas
5.6 Técnicas de prueba
5.7 Elaboración de manuales técnicos, operativos y de usuarios

CUNEZAHUALCÓYOTL

Contenido sintético



Objetivo

Identificar los conceptos básicos de la ingeniería del software.

1.0 Introducción

- La economía de todos los países desarrollados es dependiente del software.
- Actualmente cada vez mas sistemas son controlados por software.
- La Ingeniería de Software concierne a teorías, métodos y herramientas para el desarrollo profesional de software.
- El gasto en la Ingeniería de Software, representa un alto porcentaje del PIB de los países desarrollados (Pressman, 2001).

La crisis se caracterizo por los siguientes problemas (Salvador, 2012):

- Imprecisión en la planificación del proyecto y estimación de los costos.
- Baja calidad del software.
- Dificultad de mantenimiento de programas con un diseño poco estructurado, etc.

Por otra parte se exige que el software sea eficaz y barato tanto en el desarrollo como en la compra.

¿OPTIMIZAR ?

Y ésta es la razón por la cual necesitamos una computadora.

Figura 1. Característica del software

CU NEZAHUALCÓYOTL

1.1 Conceptos básicos

¿Qué es Software?

Según la IEEE *Standard Glossary of Software Engineering Terminology*" (Stad. 610,12-1990) define un software como programa de computador, procedimiento, y la documentación y los posiblemente asociados relacionados con la operación de un

CU NEZAHUALCÓYOTL

- El software puede ser desarrollado para un cliente en particular o para un mercado general.

- El software puede ser: genérico o a la medida.

- El software puede ser creado desarrollando nuevos programas, configurando sistemas de software genéricos o reutilizando los existentes.

¿Qué es la Ingeniería de Software?

Es una disciplina de la ingeniería que concierne a todos los aspectos de la producción de software.

En ese sentido los Ingenieros de software deben:

- Adoptar un enfoque sistemático para llevar a cabo su trabajo
- Utilizar las herramientas y técnicas apropiadas para resolver problemas planteados, de acuerdo a los recursos disponibles y a las restricciones de desarrollo (Bohem, 1976).

CU NEZAHUALCÓYOTL

Ingeniería de Software

- Actividad de trabajo en equipo.
- Aplicación de métodos y técnicas para racionalizar los recursos de acuerdo a planes y objetivos definidos (Meyer, 1988).

=> *Características fundamentales de la ingeniería de software: sistematicidad / disciplina/ cuantificación* <=

Sunny Auyang: " *Ingeniería es la ciencia de la producción, la cual junto a la producción es la mas fundamental de las actividades humanas*"

Ingeniería como actividad humana es la aplicación del conocimiento y los métodos científicos al diseño y la producción de procesos complejos.

CU NEZAHUALCÓYOTL

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

- ✓ La ingeniería de sistemas concierne a todos los aspectos del desarrollo de sistemas basados en cómputo incluyendo hardware, software y la ingeniería de procesos.
- ✓ La ingeniería de software es una parte de este proceso que comprende el desarrollo del software, control, aplicaciones y bases de datos del sistema.
- ✓ Los ingenieros de software son los encargados de la especificación del sistema, diseño, la integración y la puesta en marcha (Kenneth et al., 2008).

CU NEZAHUALCÓYOTL

¿Qué es un proceso de software?

Un conjunto estructurado de actividades cuya meta es el desarrollo o evolución de un software (Preesman, 2001).

Dentro de las actividades del proceso de software se enmarcan:

- Especificación
- Desarrollo
- Validación
- Evaluación (cambio/adaptación de las nuevas demandas)

CU NEZAHUALCÓYOTL

1.2 La importancia de la ingeniería del software

Se centra en los métodos, herramientas y procedimientos para establecer un control en el desarrollo del software, lo que permite construir software de calidad de forma productiva y evitando posibles errores humanos.

La calidad del software puede parecer un concepto alejado de la vida diaria de la mayoría de las personas, pero nada más lejos de la realidad.

Automatizar procesos en el desarrollo del software supone mejorar las aplicaciones, disminuir las posibles incidencias en el mismo, lo que, para las empresas va a suponer, optimizar las funcionalidades y maximizar el rendimiento de sus productos software y de su cartera de servicios.

CU NEZAHUALCÓYOTL

Figura 2. Importancia del software (Preessman, 2001)

¿Cómo lograremos llegar a cumplir el objetivo de esta ingeniería?

La respuesta es dada por la misma ingeniería de software. Ella es la responsable de facilitar los elementos para llegar a dar esa solución de software eficiente.

¿Cómo lograremos llegar a cumplir el objetivo de esta ingeniería?

La respuesta es dada por la misma ingeniería de software. Ella es la responsable de facilitar los elementos para llegar a dar esa solución de software eficiente.

CU NEZAHUALCÓYOTL

¿Cuáles son los retos de la ingeniería de software?

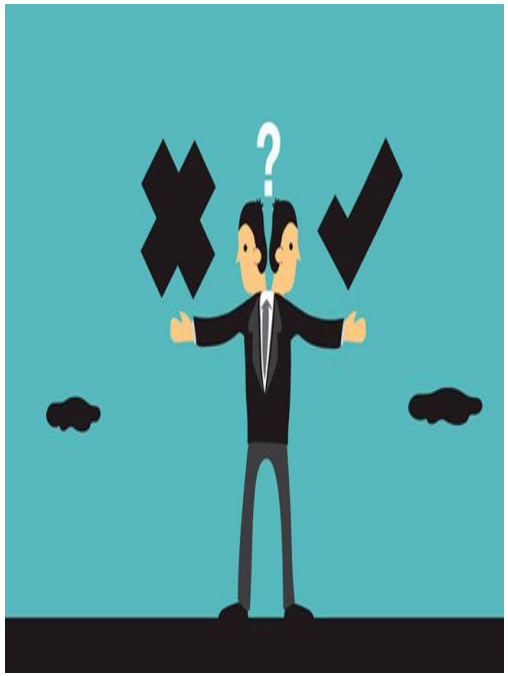
✓**Heterogeneidad.** Desarrollar técnicas para construir software que pueda hacer frente a plataformas y ambientes de ejecución heterogéneos (diferentes tipos de Hardware y Software).

✓**Tiempos de entrega**. Desarrollar técnicas que permitan reducir los tiempos de entrega del software sin comprometer la calidad.

✓**Confianza.** Desarrollar técnicas que permitan que los usuarios confíen plenamente en el software

CU NEZAHUALCÓYOTL

Responsabilidad ética y social



✓ La Ingeniería de Software comprende responsabilidades que van más allá de la simple aplicación de habilidades técnicas.

✓ Los ingenieros de Software deben actuar de manera honesta y ética si desean ser respetados como profesionales.

✓ Una conducta ética es más que sólo respetar la ley.

✓ Confidencialidad.

✓ Competencia.

✓ Derechos de propiedad intelectual.

✓ Mal uso de la computadora

CUNEZAHUALCÓYOTL

Parámetros deseables de un software (Pow, 1998)

Compatibilidad: Es la facilidad por la cual el software puede ser combinado con otro software.

Correctitud: Es el grado en que el software cumple con los requerimientos específicos, y que dichos requerimientos cumplan con las necesidades asociadas.

Corrección: Facilidad con la cual los errores latentes pueden ser encontrados y corregidos en el software.

Eficiencia: Grado en el que el software utiliza los recursos de hardware de manera efectiva. A menudo la eficiencia se sobre-enfatiza a expensas de otras metas.

Seguridad: El software se protege así mismo de accesos o modificaciones no autorizadas.

Comprensión: Facilidad con la cual los humanos pueden comprender el software y su documentación.

Uso-amigable: Facilidad con que los humanos pueden usar u operar e; software.

Validez: Es la facilidad por la cual el software puede demostrar ser correcto.

CU NEZAHUALCÓYOTL

Parámetros deseables de un software (Pow, 1998)	**Flexibilidad:** Facilidad con la que el software puede ser modificado para cumplir con cambios requeridos.
	Mantenible: Es una combinación de correcciones y flexibilidad.
	Portabilidad: Es la facilidad con la cual un software puede ser transformado de una plataforma de hardware o de software distinta.
	Confiabilidad: Grado en el que el software funciona correctamente a través del tiempo.
	Reusabilidad: El software puede ser usado para propósitos distintos del origen.
	Robustez: Grado con que el software funciona correcta mente en condiciones anormales.
	Verificación: El software puede demostrar cumplir el estándar de desarrollo de procedimientos.

CU NEZAHUALCÓYOTL

1.3 Historia de la Ingeniería del Software

- Ingeniería del Software, es el término utilizado por Fritz Bauer en la primera conferencia sobre desarrollo de software patrocinada por el Comité de Ciencia de la OTAN celebrada en Garmisch (Alemania), en octubre de 1968, previamente había sido utilizado por el holandés Edsger Dijkstra en su obra *The Humble Programmer*.

CU NEZAHUALCÓYOTL

En los comienzos:

- El programado era el usuario
- Habían pocas computadoras y muy caras
- Los problemas a resolver bien conocidos y simples.

CU NEZAHUALCÓYOTL

Su origen se debió a que el entorno de desarrollo de sistemas software adolecía de:

- Retrasos considerables en la planificación
- Poca productividad
- Elevadas cargas de mantenimiento
- Demandas cada vez más desfasadas frente a las ofertas
- Baja calidad y fiabilidad del producto
- Dependencia de los realizadores

Esto es lo que se ha denominado habitualmente "crisis del software", que históricamente se generó en los siguientes pasos:

- Primera Fase. Los albores (1945-1955)
  Programar no es una tarea diferenciada del diseño de una máquina
  Uso de lenguaje máquina y ensamblador.

- Segunda Fase. El florecimiento (1955-1965)
  Aparecen multitud de lenguajes
  Se pensaba que era posible hacer casi todo.

- Tercera Fase. La crisis (1965-1970)
  Desarrollo inacabable de grandes programas
  Ineficiencia, errores, coste impredecible
  Nada es posible.

- Cuarta Fase. Innovación conceptual (1970-1980)
  Fundamentos de programación
  Verificación de programas
  Metodologías de diseño.

- Quinta Fase. El diseño es el problema (1980-?)
  Entornos de programación
  Especificación formal
  Programación automática.

- Actualmente está surgiendo una gran expectativa ante la evolución de la Ingeniería del Software, al ir apareciendo nuevos métodos y herramientas formales que van a permitir en el futuro un planteamiento de ingeniería en el proceso de elaboración de software.

- Dicho planteamiento permitirá dar respuesta a los problemas de:
  - Administración
  - Calidad
  - Productividad
  - Fácil mantenimiento

CU NEZAHUALCÓYOTL

1.4 Descripción del ciclo de desarrollo de un producto de software

El término **ciclo de vida del software** describe el desarrollo de *software*, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para **validar** el desarrollo de la aplicación, es decir, para garantizar que el *software* cumpla los requisitos para la aplicación y **verificación** de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

CU NEZAHUALCÓYOTL

¿Qué queremos decir con proceso de desarrollo?

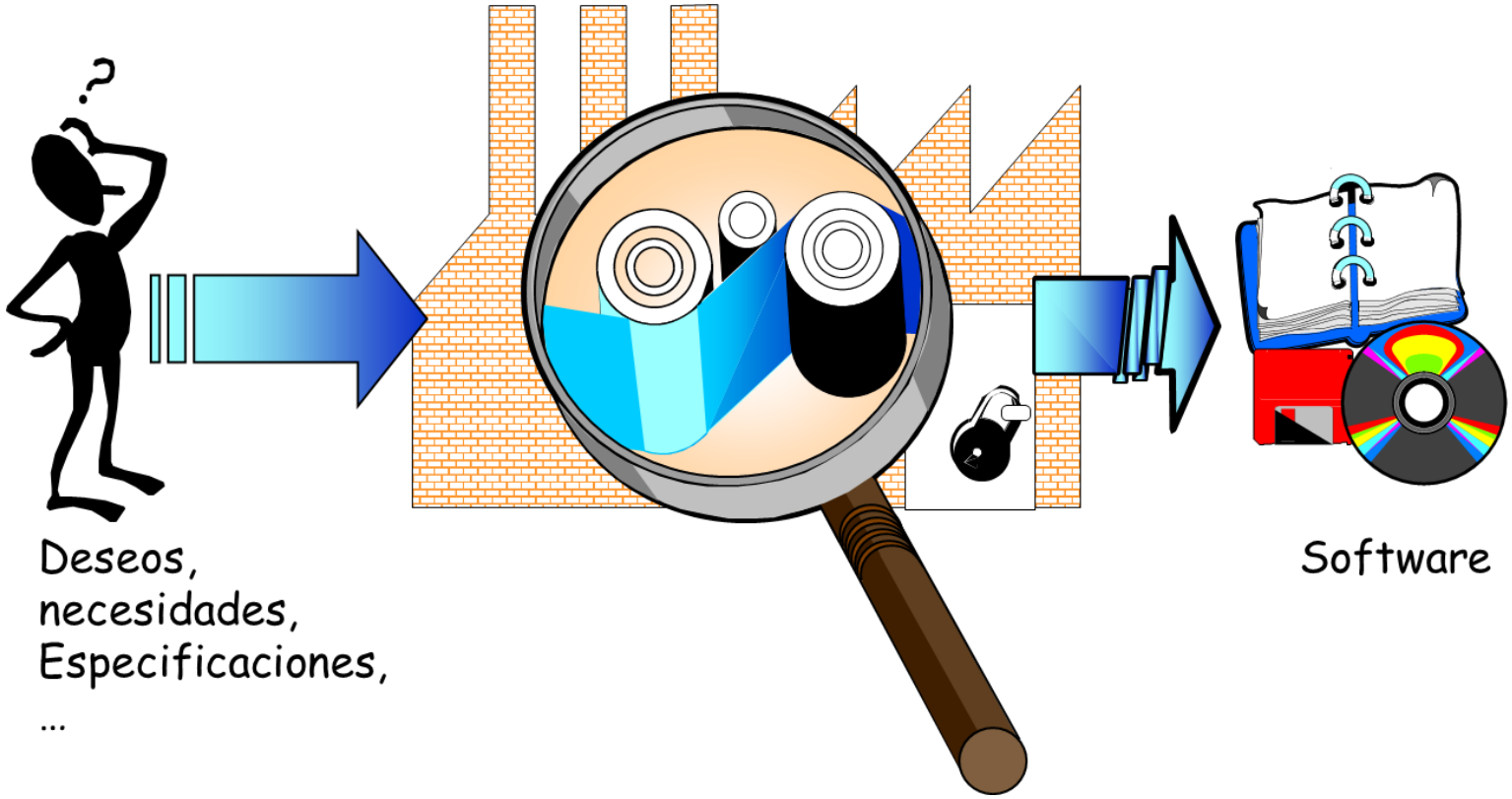


Figura 3. visión rica del proceso de software (Elaboración propia, 2018)

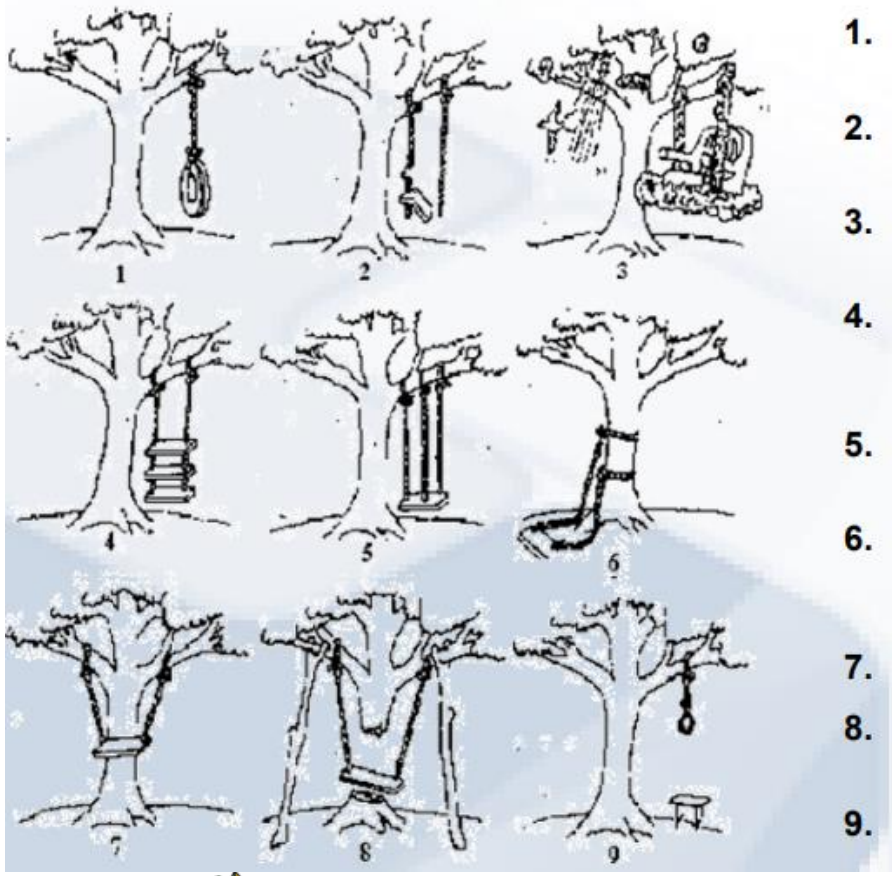
CU NEZAHUALCÓYOTL

Un proceso de desarrollo de software es la descripción de una secuencia de actividades que deben ser seguida por un equipo de trabajadores para generar un conjunto coherente de productos, uno de los cuales en el programa del sistema deseado (Braude, 2001).

El objetivo básico del proceso es hacer predecible el trabajo que se requiere:

- Predecir el costo
- Mantener un nivel de calidad
- Predecir el tiempo de desarrollo

El Típico Problema de Incomunicación



1. **Necesidad**: lo que el cliente realmente quería.
2. **Cliente**: lo que fue capaz de describir como una clara necesidad.
3. **Proceso de ventas**: lo que el fabricante de software prometió al cliente.
4. **Requisitos**: los requisitos descritos por el cliente, tal como finalmente fueron entendidos.
5. **Análisis**: la especificación formal de los requisitos realizada por los analistas.
6. **Diseño**: la especificación del funcionamiento del sistema para satisfacer los requisitos analizados.
7. **Codificación**: lo que escribió el programador.
8. **Instalación**: lo que realmente fue instalado al cliente.
9. **Pruebas**: lo que los responsables vieron en el sistema.

Figura 4. Ejemplo típico del problema de incomunicación

CU NEZAHUALCÓYOTL

1.4.1 Etapas de desarrollo

El ciclo de vida del desarrollo de software: De acuerdo con Pressman, el modelo lineal secuencial contempla cuatro fases que se deben llevar a cabo. A continuación se describen estas fases con sus respectivas actividades (Jacaboson et al., 2000).

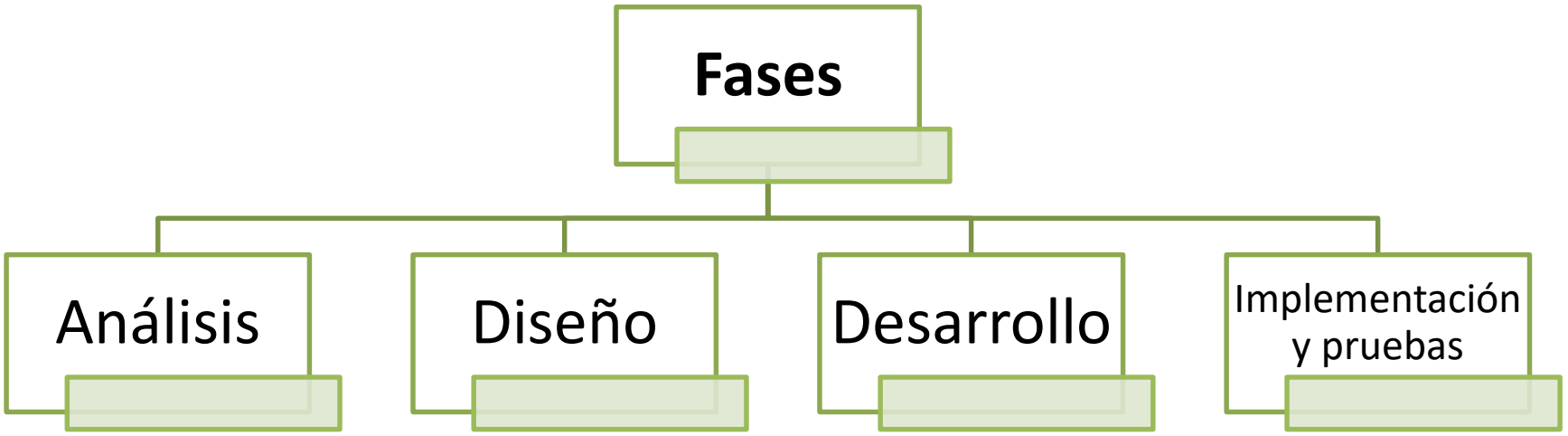


Figura 5. Fases del desarrollo del software (Jacoboson et., 2000)

CU NEZAHUALCÓYOTL

Por ello los profesionales del software se fijan en los requisitos que piden los clientes para estudiar qué requisitos están incompletos, cuales son ambiguos y cuales son simplemente contradictorios.

Para **prevenir que los requisitos que sean incorrectos**, es útil hacer **demostraciones prácticas** de cómo funcionaría la aplicación con frecuencia.

ANÁLISIS

Dentro de la fase de análisis se realiza:
Entender o conocer por completo el sistema o proceso actual;
Identificar y analizar su problemática o deficiencias y con estos elementos, determinar o proponer como se puede emplear un posible entorno computacional, en forma más optima (si es posible y necesario), para así poder efectuar los procedimientos de la empresa o área correspondiente en una forma más efectiva y eficiente.

CU NEZAHUALCÓYOTL

Entonces, el **ANÁLISIS DE SISTEMAS,** es el proceso que sirve para recopilar e interpretar los procesos y sus datos, identificar y diagnosticar problemas u oportunidades y utilizar esta información con el fin de proponer una solución a una problemática. Esta son las funciones que debe realizar un analista de sistemas.

Lo básico de la fase de análisis:

- Conocer el ayer/pasado
- Analizar, evaluar y diagnosticas el hoy
- Proponer el mañana

Todo esto dentro de un medio ambiente (contexto)

CU NEZAHUALCÓYOTL

DISEÑO

El diseño es una solución, es decir, es un traducción de los requerimientos en formas que los satisfagan.

Cuando un diseñador elabora un «diseño» escribe las especificaciones detalladas del nuevo sistema, esto es, se

CU NEZAHUALCÓYOTL

DISEÑO CONTINUACIÓN

Diseño estructurado: el arte de diseñar los componentes de un sistema y las interrelaciones entre dichos componentes en la mejor forma posible.

Según Yourdon; es el proceso de decidir cuales componentes interconectados en que forma pueden resolver un problema bien especificado.

Objetivos a cumplir en la fase de diseño

«Diseño» significa un plan para delinear la forma y el detalle de una solución. Es el proceso que determina las mejores características del sistema final, establece fronteras superiores e inferiores en el comportamiento y en la calidad que la implementación debe tener, así mismo determina el costo final que tendrá y su duración.

Al proponer el nuevo sistema se debe considerar que es[illegible] sea:

- Más eficiente.
- Fácil de mantener.
- Modificable (escalable)
- Flexible
- Útil

CU NEZAHUALCÓYOTL

DESARROLLO

Esta fase consiste en traducir el diseño en una forma legible para la maquina.

La generación de código se refiere tanto a la parte de generación de interfaz como a la parte en la cual se añadirá el comportamiento de estas interfaz.

CU NEZAHUALCÓYOTL

IMPLEMENTACIÓN Y PRUEBAS

Una vez que se ha generado el código, comienzan la implementación y pruebas del software o sistema que se ha desarrollado. De acuerdo a Pressman, el proceso de pruebas se centra en los profesos logísticos internos del software asegurando que todas las sentencias de las pruebas para la detención de errores.

Se requiere probar el software con sujetos reales que puedan evaluar el comportamiento del software con el fin de proporcionar retroalimentación a los desarrolladores

CU NEZAHUALCÓYOTL

Conclusión

- La **ingeniería de software** es una aplicación práctica del conocimiento científico para **proveer metodologías y técnicas** que ayuden a desarrollar sistemas de software a tiempo, y a su vez que aseguren que el desarrollador cumpla con las expectativas de **calidad** y permanezca dentro del **presupuesto.**

CU NEZAHUALCÓYOTL

Bibliografía

Eric Braude, Software Engineering. An Object-Oriented Perspective, John Wiley & Sons, 2001, p. 30.

Jacaboson, I., Booch, G., Rumbaugh J., El Proceso Unificado de Desarrollo de Software, Addison Wesley 2000.

Pressman Roger, Ingeniería del Software un enfoque práctico, Ed. Mc Graw-Hill, 2001.

Kenneth C. Laudon, Jane P. Laudon, Sistemas de información Gerencial, Administración de la empresa digital, Pearson Prentice Hall, 2008.

Pressman Roger, Ingeniería del Software un enfoque práctico, Ed. Mc Graw-Hill, 2001.

Salvador Sánchez Alonso, Miguel Ángel Sicilia Urbán, Ingeniería del Software, un enfoque desde la guía SWEBOK, Alfaomega Editores, S.A. de C.V., México, 2012.

CU NEZAHUALCÓYOTL

GRACIAS POR SU ATENCIÓN