IES INSTITUTO DE ESTUDIOS SUPERIORES

UNIDAD I

Temas desarrollados

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Desarrollo de software profesional
  - ¿Qué es lo que se entiende por ingeniería de software.?
- Ética en la ingeniería de software
  - Una breve introducción sobre los conflictos éticos que afectan a la ingeniería de software.

Antes de empezar… ¿Qué es "el software"?

Muchas personas asocian el término ***software con los programas de computadora.*** Sin embargo, tomamos una definición más amplia donde el software no son sólo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. Por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, la documentación para el usuario que explica cómo utilizar el sistema y sitios web que permitan a los usuarios descargar la información de productos recientes.

Ingenieria de software

✧La economía de todos los países desarrollados dependen del software.
✧Cada vez mas sistemas son controlados por software.
✧La ingeniería de software aplica teorías, métodos y herramientas para el desarrollo de software profesional.
✧El gasto en software representa una fracción significativa del PIB en todos los países desarrollados.

Costos del software

✧Costos de software a menudo dominan los costes del sistema informático. El costo del software en una PC suele ser mayores que el costo del hardware.
✧El mantenimiento del software cuesta mas que el costo del desarrollo del mismo. Para sistemas que tienen una larga vida, los costos de mantenimiento superan ampliamente los costos de desarrollo.
✧La ingeniería de software tiene que ver con el desarrollo de software rentable.

Productos de software

✧Productos genéricos
- Sistemas independientes que se comercializan y venden a cualquier cliente que desee comprar.
- Ejemplos - Software para PC tales como programas de gráficos, herramientas de gestión de proyectos; Software CAD; software para mercados específicos, tales como los sistemas de citas para los dentistas.

✧Productos personalizados
- Software que esté encargado por un cliente específico para satisfacer sus propias necesidades.
- Ejemplos - incorporado sistemas de control, software de control del tráfico aéreo, sistemas de monitorización de tráfico.

Especificaciones del producto

- Productos Genéricos
  - La especificación de lo que el software debe hacer es propiedad del desarrollador del software y las decisiones sobre los cambios en el software son hechas por el desarrollador.
- Productos personalizados
  - La especificación de lo que el software debe hacer es propiedad del cliente del software y él es el que toma decisiones sobre los cambios de software necesarios.

Software de Calidad

Cuando se habla de la calidad del software profesional, se debe considerar que el software lo usan y cambian personas, además de sus desarrolladores.

En consecuencia, la calidad no tiene que ver sólo con lo que hace el software.

En cambio, debe incluir el comportamiento del software mientras se ejecuta, y la estructura y organización de los programas del sistema y la documentación asociada. Esto se refleja en los llamados calidad o atributos no funcionales del software.

Ejemplos de dichos atributos son el tiempo de respuesta del software ante la duda de un usuario y la comprensibilidad del código del programa

Software de Calidad

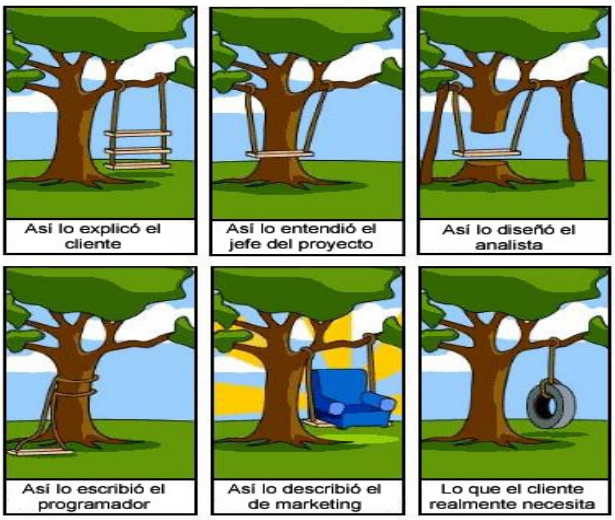
El conjunto específico de atributos que se espera de un sistema de software depende evidentemente de su aplicación. Así, un sistema bancario debe ser seguro, un juego interactivo debe tener capacidad de respuesta, un sistema de conmutación telefónica debe ser confiable, etcétera.

Veamos unos atributos que se consideran esenciales para un sistema….

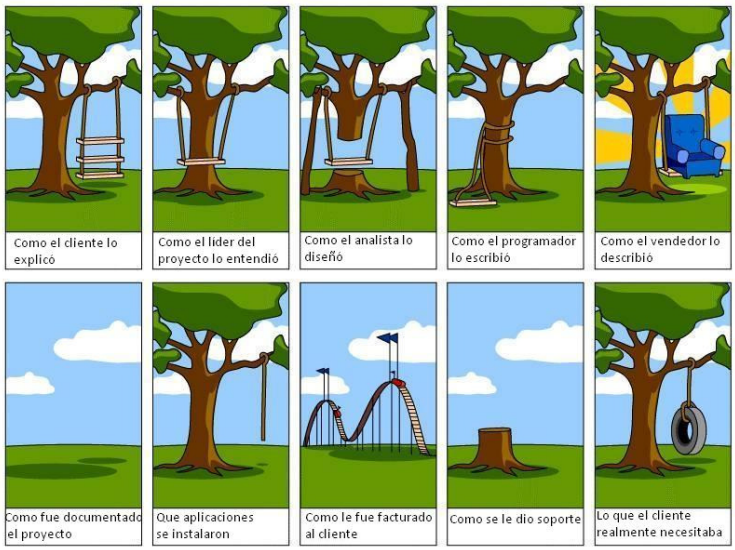
Atributos esenciales de un buen software

Características del producto	Descripción
Mantenimiento	El software debe escribirse de tal forma que pueda evolucionar para satisfacer las necesidades cambiantes de los clientes. Éste es un atributo crítico porque el cambio del software es un requerimiento inevitable de un entorno empresarial variable.
Confiabilidad y seguridad	La confiabilidad del software incluye una variedad de características incluyendo confiabilidad, protección y seguridad. El software fiable no debe causar daño físico o económico en caso de fallo del sistema. Los usuarios malintencionados no deben poder acceder o dañar el sistema.
Eficiencia	El software no debe desperdiciar el uso de los recursos del sistema, como los ciclos de memoria y procesador. Por lo tanto, la eficiencia incluye la capacidad de respuesta, el tiempo de procesamiento, la utilización de la memoria, etc.
Aceptabilidad	El software debe ser aceptable para el tipo de usuario para el que está diseñado. Esto significa que debe ser comprensible, utilizable y compatible con otros sistemas que utilizan.

El columpio



El columpio II



Ingeniería de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧La ingeniería de software es una disciplina de la ingeniería que se ocupa de todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema hasta el el mantenimiento del sistema después de que haya entrado en uso.

✧Disciplina de Ingeniería

- El uso de las teorías y los métodos adecuados para resolver los problemas teniendo en cuenta las limitaciones financieras y de organización.

✧Todos los aspectos de la producción de software

- No sólo el proceso técnico de desarrollo. También la gestión de proyectos y el desarrollo de herramientas, métodos, etc., para apoyar la producción de software.

Importancia de la ingeniería de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧Cada vez más personas y la sociedad en general dependen de sistemas de software avanzados. Tenemos que ser capaces de producir sistemas fiables y de confianza económica y rápida.

✧Por lo general es más barato, en el largo plazo, el uso de métodos de ingeniería de software y técnicas para los sistemas de software en lugar de escribir los programas como si fuera un proyecto de programación personal. Para la mayoría de tipos de sistema, la mayoría de los costos son los costos de cambiar el software después de que ha entrado en uso.

Preguntas planteadas con frecuencia sobre la ingeniería de software

¿Que es software?	Programas de cómputo y documentación asociada. Los productos de software se desarrollan para un cliente en particular o para un mercado en general.
¿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.
¿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.
¿Cuáles son las actividades fundamentales de la ingeniería de software?	Especificación, desarrollo, validación y evolución del 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.
¿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.

Preguntas planteadas con frecuencia sobre la ingeniería 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.
¿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.
¿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.

Actividades del proceso de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧Especificación del software, donde clientes e ingenieros definen el software que se producirá y las restricciones en su operación.

✧ Desarrollo del software, donde se diseña y programa el software.

✧Validación del software, donde se verifica el software para asegurar que sea lo que el cliente requiere.

✧Evolución del software, donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.

Detalles generales que afectan la mayoría del software

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧Heterogeneidad
- Cada vez con mayor frecuencia se requieren sistemas que operen como sistemas distribuidos a través de redes que incluyan diferentes tipos de computadoras y dispositivos móviles.

✧Cambio empresarial y social
- Los negocios y la sociedad cambian de manera rápida, conforme se desarrollan las economías emergentes y nuevas tecnologías están a la disposición. Ambos necesitan tener la posibilidad de cambiar su software existente y desarrollar rápidamente uno nuevo.

✧Seguridad y confianza
- Dado que el software está vinculado con todos los aspectos de la vida, es esencial confiar en dicho software

Diversidad de ingeniería de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧Hay muchos tipos diferentes de sistemas de software y no existe un conjunto universal de las técnicas de software que es aplicable a todas ellas.

✧Los métodos de ingeniería de software y las herramientas que se utilizan dependen del tipo de aplicación que se está desarrollando, los requisitos del cliente y los antecedentes del equipo de desarrollo.

Tipos de aplicaciones

IES INSTITUTO DE ESTUDIOS SUPERIORES

✧Aplicaciones autónomas

- Estos son los sistemas de aplicación que se ejecutan en un equipo local, como un PC. Incluyen toda la funcionalidad necesaria y no es necesario estar conectado a una red. Ejemplos de tales aplicaciones son las de oficina en una PC, programas CAD, software de manipulación de fotografías, etcétera.

✧Aplicaciones basadas en transacciones interactivas

- Las aplicaciones que se ejecutan en un equipo remoto y se puede acceder por los usuarios desde sus propios ordenadores o terminales. Esto incluye aplicaciones web como aplicaciones de comercio electrónico.

Tipos de aplicaciones

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Sistemas de procesamiento por lotes
  - Estos son sistemas de negocios que están diseñados para procesar los datos en grandes lotes. Procesan un gran número de entradas individuales para crear salidas correspondientes. Los ejemplos de sistemas batch incluyen sistemas de facturación periódica, como los sistemas de facturación telefónica y los sistemas de pago de salario.

- Sistemas de entretenimiento
  - Se trata de sistemas que son principalmente para su uso personal y que están destinados a entretener al usuario.

Tipos de aplicaciones

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Sistemas de control incrustados
  - Se trata de sistemas de control de software que controlan y gestionan los dispositivos de hardware. Numéricamente, hay probablemente más sistemas integrados que cualquier otro tipo de sistema. Ejemplos de sistemas embebidos incluyen el software en un teléfono móvil (celular), el software que controla los frenos antibloqueo de un automóvil y el software en un horno de microondas para controlar el proceso de cocinado

- Sistemas de recopilación de datos
  - Se trata de sistemas que recopilan datos de su entorno utilizando un conjunto de sensores y envían los datos a otros sistemas para el procesamiento. El software tiene que interactuar con los sensores y se instala regularmente en un ambiente hostil, como en el interior de un motor o en una ubicación remota.

Tipos de aplicaciones

- Sistemas para el modelado y simulación
  - Éstos son sistemas que desarrollan científicos e ingenieros para modelar procesos o situaciones físicas, que incluyen muchos objetos separados interactuantes. Son computacionalmente intensivos y para su ejecución requieren sistemas paralelos de alto desempeño.

- Sistemas de sistemas
  - Estos son sistemas que están compuestos de un número de otros sistemas de software. Por ejemplo: producto del software genérico, como un programa de hoja de cálculo.

Fundamentos de la ingenieria de software

- Algunos principios fundamentales se aplican a todos los tipos de sistema de software, con independencia de las técnicas de desarrollo utilizados:
  - Los sistemas deben ser desarrollados mediante un proceso de desarrollo dirigido y entendido. Por supuesto, diferentes procesos se utilizan para diferentes tipos de software.
  - La fiabilidad y el rendimiento son importantes para todos los tipos de sistema.
  - La comprensión y la gestión de la especificación de requisitos de software y (lo que el software debe hacer) son importantes.
  - En su caso, debe volver a utilizar el software que ya ha sido desarrollado en lugar de escribir un nuevo software.

La ingeniería de software y la web

✧La Web es ahora una plataforma para ejecutar aplicaciones y las organizaciones están desarrollando cada vez más los sistemas basados en la web en lugar de los sistemas locales.
✧Los servicios Web permiten la funcionalidad de la aplicación para acceder a través de Internet.
✧La computación en nube es un enfoque para la prestación de servicios de informática donde las aplicaciones se ejecutan de forma remota en la "nube".
▪Los usuarios no compran software de pago de compra en función del uso.

Ingenieria software web

✧Reutilización de software es el enfoque dominante para la construcción de sistemas basados en la web.
- ▪Durante la construcción de estos sistemas, se piensa en como puede ser construido a partir de componentes y sistemas de software pre-existentes.

✧Los sistemas basados en la Web deben ser desarrollados y entregados de forma incremental.
- ▪En la actualidad se reconoce en general que no es práctico para especificar todos los requisitos para este tipo de sistemas de anticipación..

✧Las interfaces de usuario están limitadas por las capacidades de los navegadores web.
- ▪Las tecnologías como AJAX permiten interfaces enriquecidas que se crean dentro de un navegador web, pero siguen siendo difíciles de usar. Formularios Web con scripts locales son más comúnmente utilizados.

Ingeniería de software basada en la Web

✧Los sistemas basados en la Web son sistemas distribuidos complejos, pero los principios fundamentales de la ingeniería de software previamente discutidos son tan aplicables a ellos como lo son para cualquier otro tipo de sistema.
✧Las ideas fundamentales de la ingeniería de software, que se analizan en la sección anterior, se aplican al software basado en la web de la misma manera que se aplican a otros tipos de sistemas de software.

¿Cuáles son los retos fundamentales que afronta la ingeniería del software?

En el siglo XXI, la ingeniería del software afronta tres retos fundamentales:

1. ***El reto de la heterogeneidad***. Cada vez más, se requiere que los sistemas operen como sistemas distribuidos en redes que incluyen diferentes tipos de computadoras y con diferentes clases de sistemas de soporte. A menudo es necesario integrar software nuevo con sistemas heredados más viejos escritos en diferentes lenguajes de programación. El reto de la heterogeneidad es desarrollar técnicas para construir software confiable que sea lo suficientemente flexible para adecuarse a esta heterogeneidad.

¿Cuáles son los retos fundamentales que afronta la ingeniería del software?

IES INSTITUTO DE ESTUDIOS SUPERIORES

2. ***El reto de la entrega.*** Muchas técnicas tradicionales de ingeniería del software consumen tiempo. El tiempo que éstas consumen es para producir un software de calidad. Sin embargo, los negocios de hoy en día deben tener una gran capacidad de respuesta y cambiar con mucha rapidez. Su software de soporte también debe cambiar con la misma rapidez. El reto de la entrega es reducir los tiempos de entrega para sistemas grandes y complejos sin comprometer la calidad del sistema.

¿Cuáles son los retos fundamentales que afronta la ingeniería del software?

IES INSTITUTO DE ESTUDIOS SUPERIORES

3. ***El reto de la confianza.*** Puesto que el software tiene relación con todos los aspectos de nuestra vida, es esencial que podamos confiar en él. Esto es especialmente importante en sistemas remotos de software a los que se accede a través de páginas web o de interfaces de servicios web, El reto de la confianza es desarrollar técnicas que demuestren que los usuarios pueden confiar en el software.

Por supuesto, éstos no son independientes. Por ejemplo, es necesario hacer cambios rápidos a los sistemas heredados para proveerlos de una interfaz de servicio web. Para tratar estos retos, necesitaremos nuevas herramientas y técnicas, así como formas innovadoras de combinación y uso de métodos de ingeniería del software existentes.

¿Qué es un proceso de software?

IES INSTITUTO DE ESTUDIOS SUPERIORES

• **Un Proceso** es Un conjunto de actividades interrelacionadas que transforman entradas en salidas. (ISO 12207/UNE 77104)

• **Un Proceso Software (PS)** es un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software. (Fugetta, 2000)

¿Qué es un proceso de software?

IES INSTITUTO DE ESTUDIOS SUPERIORES

Un proceso del software es un conjunto de actividades y resultados asociados que producen un producto de software. Estas actividades son llevadas a cabo por los ingenieros de software.

Existen cuatro actividades fundamentales de procesos que son comunes para todos los procesos del software. Estas actividades son:

1. ***Especificación del software*** donde los clientes e ingenieros definen el software a producir y las restricciones sobre su operación.
2. ***Desarrollo del software*** donde el software se diseña y programa.
3. ***Validación del software*** donde el software se valida para asegurar que es lo que el cliente requiere.
4. ***Evolución del software*** *donde el software se modifica para adaptarlo a los cambios requeridos* por el cliente y el mercado.

¿Qué es un proceso de software?

Diferentes tipos de sistemas necesitan diferentes procesos de desarrollo.

Por ejemplo, el software de tiempo real en un avión tiene que ser completamente especificado antes de que empiece el desarrollo, mientras que en un sistema de comercio electrónico, la especificación y el programa normalmente son desarrollados juntos.

Por lo tanto, estas actividades genéricas pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de software. Sin embargo, el uso de un proceso inadecuado del software puede reducir la calidad o la utilidad del producto de software que se va a desarrollar y/o incrementar los costes de desarrollo.

¿Qué es un modelo de procesos del software?

Un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso. Estos modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software.

¿Qué es un modelo de procesos del software?

Algunos ejemplos de estos tipos de modelos que se pueden producir son:

- ***Un modelo de flujo de trabajo.*** Muestra la secuencia de actividades en el proceso junto con sus entradas, salidas y dependencias. Las actividades en este modelo representan acciones humanas.
- ***Un modelo de flujo de datos o de actividad.*** Representa el proceso como un conjunto de actividades, cada una de las cuales realiza alguna transformación en los datos. Muestra cómo la entrada en el proceso, tal como una especificación, se transforma en una salida, tal como un diseño. Pueden representar transformaciones llevadas a cabo por las personas o por las computadoras.
- ***Un modelo de rol/acción.*** Representa los roles de las personas involucrada en el proceso del software y las actividades de las que son responsables.

¿Qué es un modelo de procesos del software?

La mayor parte de los modelos de procesos del software se basan en uno de los tres modelos generales o paradigmas de desarrollo de software:

- ***El enfoque en cascada.*** Considera las actividades anteriores y las representa como fases de procesos separados, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera. Después de que cada etapa queda definida «se firma» y el desarrollo continúa con la siguiente etapa.

- ***Desarrollo iterativo.*** Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Este se refina basándose en las peticiones del cliente para producir un sistema que satisfaga las necesidades de dicho cliente. El sistema puede entonces ser entregado. De forma alternativa, se puede reimplementar utilizando un enfoque más estructurado para producir un sistema más sólido y mantenible.

¿Qué es un modelo de procesos del software?

- ***Ingeniería del software basada en componentes (CBSE).*** Esta técnica supone que las partes del sistema existen. El proceso de desarrollo del sistema se enfoca en la integración de estas partes más que desarrollarlas desde el principio.

CASE

CASE (Ingeniería del Software Asistida por Computadora) comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso del software, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas.

En la actualidad, todos los métodos vienen con tecnología CASE asociada, como los editores para las notaciones utilizadas en el método, módulos de análisis que verifican el modelo del sistema según las reglas del método y generadores de informes que ayudan a crear la documentación del sistema. Las herramientas CASE también incluyen un generador de código que automáticamente genera código fuente a partir del modelo del sistema y de algunas guías de procesos para los ingenieros de software.

CASE

Las herramientas de desarrollo del software (llamadas en ocasiones herramientas de Ingeniería de Software Asistido por Computadora o CASE, por las siglas de *Computer-Aided Software Engineering) son* programas usados para apoyar las actividades del proceso de la ingeniería de software. En consecuencia, estas herramientas incluyen editores de diseño, diccionarios de datos, compiladores, depuradores (*debuggers),* herramientas de construcción de sistema, etcétera. Las herramientas de software ofrecen apoyo de proceso al automatizar algunas actividades del proceso y brindar información sobre el software que se desarrolla.

CASE

Los ejemplos de actividades susceptibles de automatizarse son:

- Desarrollo de modelos de sistemas gráficos, como parte de la especificación de requerimientos o del diseño del software.
- Generación de código a partir de dichos modelos de sistemas gráficos.
- Producción de interfaces de usuario a partir de una descripción de interfaz gráfica, creada por el usuario de manera interactiva.
- Depuración del programa mediante el suministro de información sobre un programa que se ejecuta.
- Traducción automatizada de programas escritos, usando una versión anterior de un lenguaje de programación para tener una versión más reciente.

Puntos clave

✧La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software.
✧Los atributos esenciales de los productos de software son mantenimiento, confiabilidad, seguridad, eficiencia y aceptabilidad.
✧Las actividades de alto nivel de especificación, desarrollo, validación y evolución son parte de todos los procesos de software.

Puntos clave

✧Existen muchos tipos diferentes de sistemas y cada uno requiere para su desarrollo de herramientas y técnicas adecuadas de ingeniería de software.

✧Las ideas fundamentales de la ingeniería de software son aplicables a todos los tipos de sistemas de software.

✧Las nociones fundamentales de la ingeniería de software son universalmente aplicables a todos los tipos de desarrollo de sistema.

Procesos de Software

Objetivos

- Introducir modelos de procesos de software
- Describir tres modelos de procesos genéricos y cuándo pueden utilizarse
- Describir modelos de procesos de ingeniería de requerimientos, de desarrollo de software, pruebas y la evolución
- Explicar el modelo de Proceso Unificado de Rational
- Introducir la tecnología CASE para apoyar las actividades de proceso del software

Tópicos expuestos

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Modelos de procesos de software
- Iteración de procesos
- Actividades del proceso
- El Proceso Unificado de Rational
- Ingeniería de Software Asistida por Computadora

El proceso del software

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Un conjunto estructurado de actividades necesarias para desarrollar un Sistema de software

  -Especificación;
  -Diseño;
  -Validación;
  -Evolución.

- Un modelo de proceso del software es una representación abstracta de un proceso. Presenta una descripción de un proceso desde una perspectiva particular.

Modelos genéricos de proceso de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

•El modelo de cascada: Separadas y distintas fases de la especificación y el desarrollo.

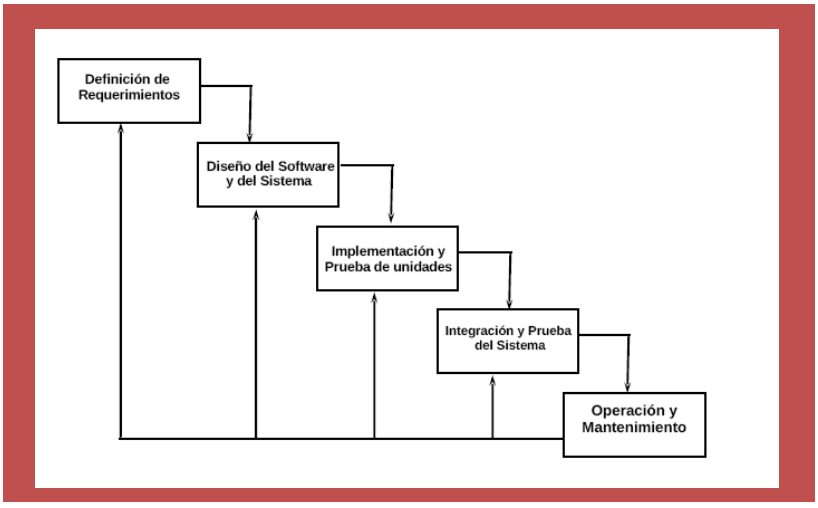
•Desarrollo evolutivo: Especificación, desarrollo y validación están intercalados.

•Ingeniería de Software Basada en Componentes: El sistema está montado a partir de los componentes existentes.

•Hay muchas variantes de estos modelos, por ejemplo, el desarrollo formal donde se toma un proceso similar al de cascada, pero la especificación es una especificación formal que se perfecciona a través de varias etapas hacia un diseño implementable.

Modelo de cascada

IES INSTITUTO DE ESTUDIOS SUPERIORES



Modelo de cascada - fases

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Análisis de requerimientos y definición
- Diseño del sistema y del software
- Ejecución y unidad de pruebas
- Implementación y pruebas del sistema
- Operación y mantenimiento
- El principal inconveniente del modelo de cascada es la dificultad de acomodar el cambio después de que el proceso está en marcha. Una fase tiene que ser completada antes de pasar a la siguiente fase.

Modelo de cascada - problemas

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Rígida división del proyecto en fases distintas que hace difícil responder a la evolución de las necesidades del cliente.
- Por lo tanto, este modelo sólo es apropiado cuando los requisitos son bien entendidos y los cambios serán bastante limitados durante el proceso de diseño.
- Pocos sistemas de negocio tienen requerimientos estables.
- El modelo de cascada se utiliza para grandes proyectos de ingeniería de sistemas donde el desarrollo de un sistema se efectúe en varios sitios.

Desarrollo evolutivo – 2 Tipos

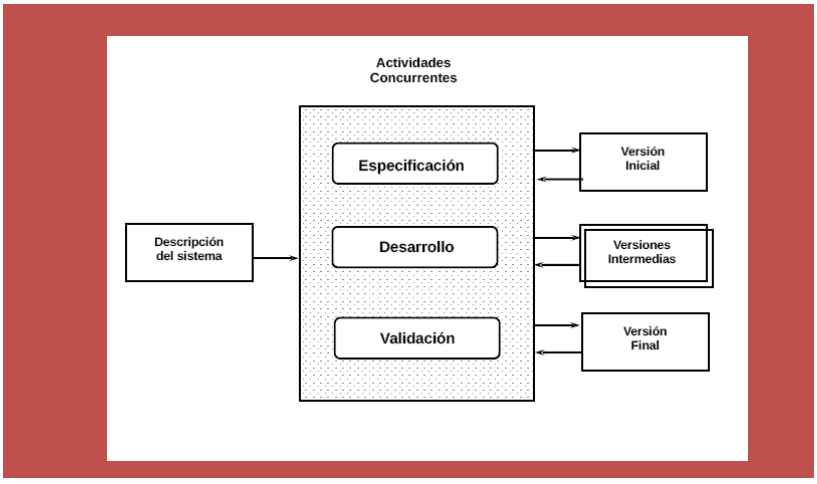
IES INSTITUTO DE ESTUDIOS SUPERIORES

•Desarrollo exploratorio
El objetivo es trabajar con los clientes y evolucionar a un sistema final a partir de un esbozo inicial de especificación. Debería empezar con buen entendimiento de los requerimientos y añadir nuevas funciones en la forma propuesta por el cliente.

•Prototipado desechable
El objetivo es comprender los requisitos del sistema. Se puede empezar con especificaciones poco entendidas.

Desarrollo evolutivo

IES INSTITUTO DE ESTUDIOS SUPERIORES



Desarrollo evolutivo

Problemas
•Poca visibilidad en el proceso;
•Los sistemas son a menudo mal estructurado;
•Habilidades especiales (por ejemplo, lenguajes para prototipado rápido) pueden ser requeridas.

Aplicabilidad
•Para sistemas interactivos pequeños o medianos;
•Para partes de grandes sistemas (por ejemplo, la interfaz de usuario);
•Para sistemas de corta vida.

Desarrollo evolutivo

Para **sistemas pequeños y de tamaño medio** (hasta 500.000 líneas de código), el enfoque evolutivo de desarrollo es el mejor. Los problemas del desarrollo evolutivo se hacen particularmente agudos para sistemas grandes y complejos con un periodo de vida largo, donde diferentes equipos desarrollan distintas partes del sistema. Es difícil establecer una arquitectura del sistema estable usando este enfoque, el cual hace difícil integrar las contribuciones de los equipos.

Para **sistemas grandes**, se recomienda un proceso mixto que incorpore las mejores características del modelo en cascada y del desarrollo evolutivo. Esto puede implicar desarrollar un prototipo desechable utilizando un enfoque evolutivo para resolver incertidumbres en la especificación del sistema. Puede entonces reimplementarse utilizando un enfoque más estructurado.
Las partes del sistema bien comprendidas se pueden especificar y desarrollar utilizando un proceso basado en el modelo en cascada. Las otras partes del sistema, como la interfaz de usuario, que son difíciles de especificar por adelantado, se deben desarrollar siempre utilizando un enfoque de programación exploratoria

Ingeniería de Software Basada en Componentes

IES INSTITUTO DE ESTUDIOS SUPERIORES

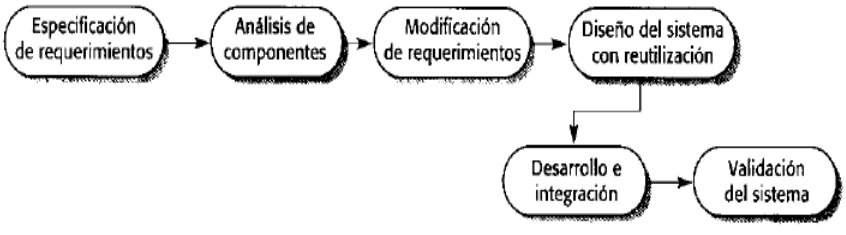
•Sobre la base de la reutilización sistemática donde se integran los sistemas de los componentes existentes o COTS (Comercial-off-the-shelf) de sistemas.

•Las fases del proceso
- -Análisis de componentes;
- -Modificación de requerimientos;
- -Diseño del sistema con reutilización;
- -Desarrollo e integración.

•Este enfoque está siendo cada vez más utilizado a medida que los componentes estándar van surgiendo.

Desarrollo orientado a la reutilización

IES INSTITUTO DE ESTUDIOS SUPERIORES



Iteración de Procesos

IES INSTITUTO DE ESTUDIOS SUPERIORES

•Los requerimientos del sistema SIEMPRE evolucionan en el transcurso de un proyecto, así que el proceso de iteración donde las fases son revisadas, siempre son parte del proceso de grandes sistemas.

•Iteración se puede aplicar a cualquiera de los modelos de procesos genéricos.

•Dos enfoques (relacionados)
- -Entrega incremental;
- -Desarrollo en Espiral.

Entrega Incremental

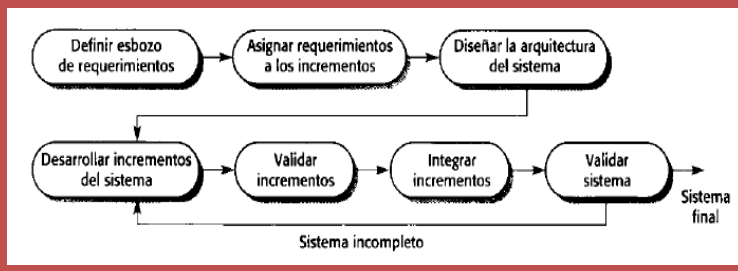
IES INSTITUTO DE ESTUDIOS SUPERIORES

•En lugar de entregar el sistema en una sola entrega, el desarrollo y la prestación se divide en incrementos, y con cada incremento se entrega parte de la funcionalidad requerida.

•Son priorizados los requerimientos de los usuarios y los requerimientos de más alta prioridad son incluidos en los primeros incrementos.

•Una vez que el desarrollo de un incremento se ha iniciado, los requisitos están congelados, a pesar de que los requerimientos de los incrementos posteriores pueden seguir evolucionando.

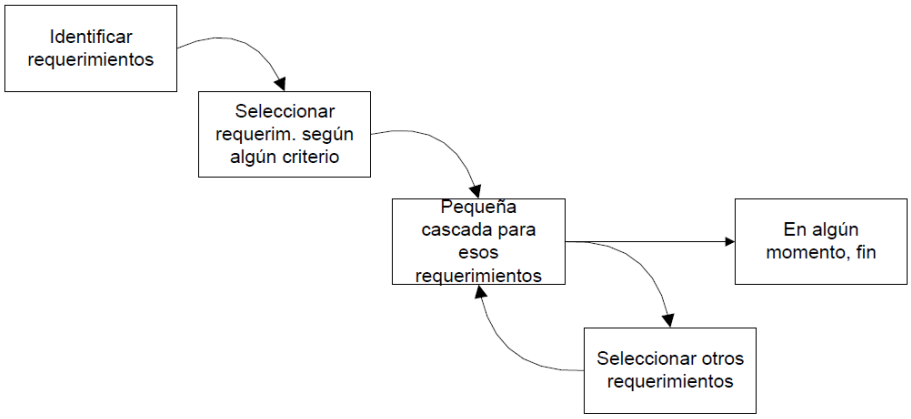
Entrega Incremental



Ventajas del Desarrollo incremental

- Entregas con valor para el cliente pueden ser hechas con cada incremento, de manera que la funcionalidad del sistema está disponible tempranamente.
- Los primeros incrementos actúan como un prototipo para ayudar a obtener requerimientos para incrementos posteriores.
- Menor riesgo de fracaso del proyecto general.
- La más alta prioridad del sistema de servicios tiende a recibir más pruebas.

Modelos iterativos e incrementales (o evolutivos)



Modelos iterativos e incrementales (o evolutivos)

Principales ventajas

-El usuario ve algo rápidamente

-Se admite que lo que se está construyendo es el sistema, y por lo tanto se piensa en su calidad desde el principio

-Se pueden atacar los principales riesgos

-Los ciclos van mejorando con las experiencias de los anteriores

Principales variaciones

-Duración de las iteraciones

-Existencia o no de tiposde iteraciones

-Cantidad de esfuerzo dedicado a identificación inicial de requerimientos

-Actitud “defensiva” ante los cambios

Desarrollo en Espiral

IES INSTITUTO DE ESTUDIOS SUPERIORES

•El proceso se representa como una espiral y no como una secuencia de actividades con marcha atrás.

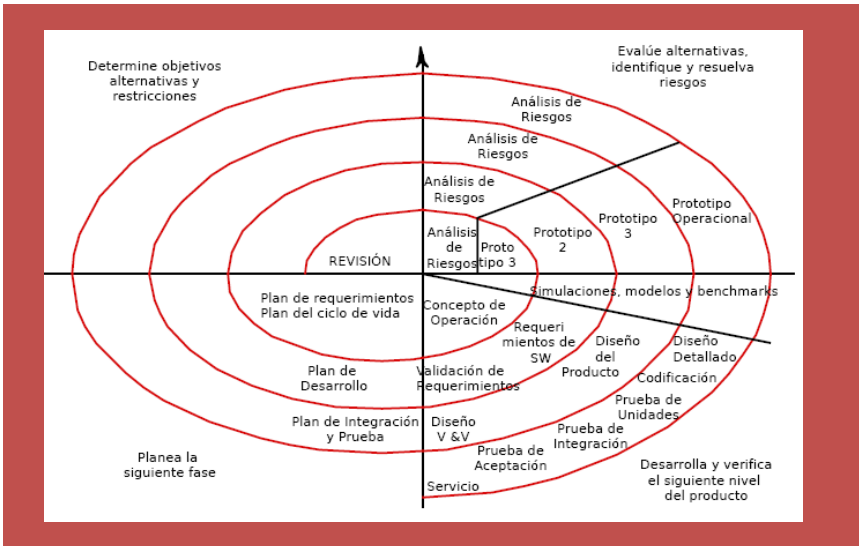
•Cada bucle en la espiral representa una fase en el proceso.

•Fases no fijas, tales como las fases de diseño o especificación - bucles de la espiral se eligen en función de lo que se necesita.

•Riesgos se evalúan de forma explícita y se resuelven a través todo el proceso.

Espiral modelo el proceso de software

IES INSTITUTO DE ESTUDIOS SUPERIORES



Sectores del modelo en espiral

•La fijación de objetivos
-Los objetivos específicos de cada fase son identificados.

•Evaluación de riesgos y su reducción
-Se evalúan los riesgos y las actividades puestas en marcha para reducir los principales riesgos.

•Desarrollo y validación
-Un modelo de desarrollo para el sistema elegido es el que puede ser cualquiera de los modelos genéricos.

•Planificación
-El proyecto es revisado y la próxima fase de la espiral es planeada.

Actividades del proceso

- Especificación de software
- Diseño e implementación de software
- Validación de software
- Evolución del software

Especificación de software

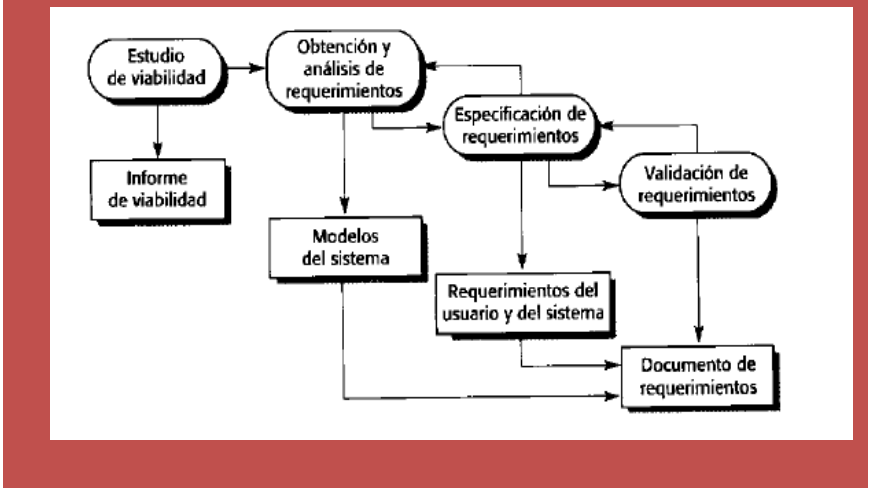
IES INSTITUTO DE ESTUDIOS SUPERIORES

•El proceso de establecer qué servicios son requeridos y las limitaciones en el funcionamiento del sistema y el desarrollo.

•Proceso de ingeniería de requerimientos
- -Estudio de viabilidad;
- -Obtención y análisis de requerimientos;
- -Especificación de los requerimientos;
- -Validación de requerimientos.

Proceso de ingeniería de requerimientos

IES INSTITUTO DE ESTUDIOS SUPERIORES



Actividades del proceso

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Especificación de software
- **Diseño e implementación de software**
- Validación de software
- Evolución del software

Diseño e implementación de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

•El proceso de convertir la especificación del sistema en un sistema ejecutable.

•Diseño de software
Diseñar una estructura de software que dé constancia de la especificación;

•Implementación
-Traducir esta estructura en un programa ejecutable;

•Las actividades de diseño e implementación están estrechamente relacionadas y pueden ser interpoladas.

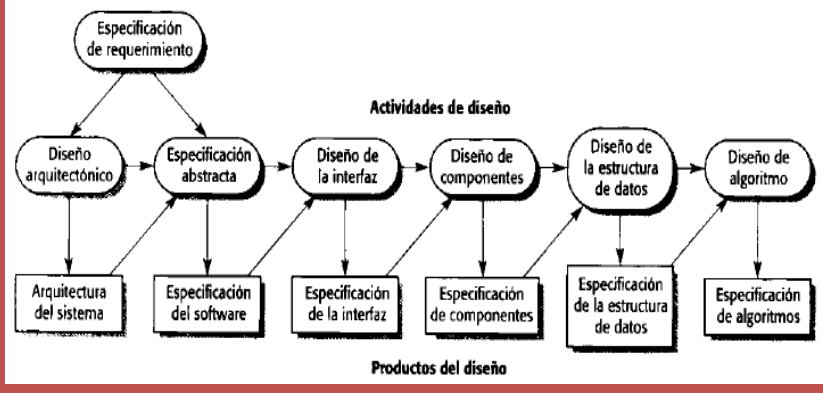
Actividades del proceso de Diseño

IES INSTITUTO DE ESTUDIOS SUPERIORES

- Diseño arquitectónico
- Especificación abstracta
- Diseño de la interfaz
- Diseño de componentes
- Diseño de estructura de datos
- Diseño de algoritmos

El proceso de diseño de software

IES INSTITUTO DE ESTUDIOS SUPERIORES



Métodos estructurados

•Enfoques sistemáticos para el desarrollo de un diseño de software.

•El diseño es por lo general documentado como un conjunto de modelos gráficos.

•Posibles modelos
- -Modelo de objetos;
- -Modelo de secuencia;
- -Modelo de transición de estados;
- -Modelo estructural;
- -Modelo de flujo de datos.

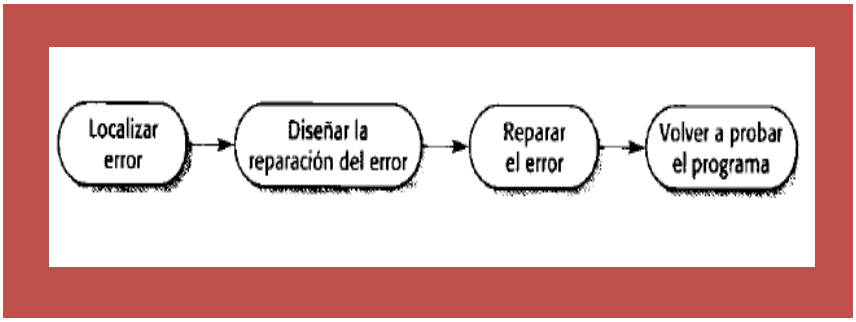
Programación y depuración

•Traducción de un diseño en un programa y la eliminación de errores de ese programa.

•La programación es una actividad personal - no hay ningún proceso de programación genérico.

•Los programadores deben acarrear alguna prueba del programa para descubrir defectos en el programa y eliminar esas fallas en el proceso de depuración.

El proceso de depuración



Actividades del proceso

- Especificación de software
- Diseño e implementación de software
- **Validación de software**
- Evolución del software

Validación de software

IES INSTITUTO DE ESTUDIOS SUPERIORES

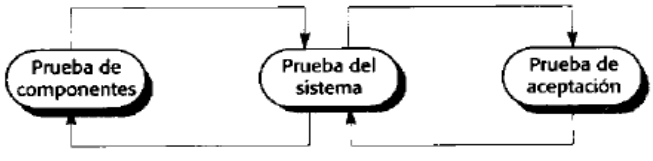
•Verificación y validación (V & V) es para poner de manifiesto que un sistema cumple con su especificación y cumple los requerimientos del cliente.

•Implica el control y revisión de los procesos y pruebas del sistema.

•La prueba del sistema implica la ejecución del sistema con casos de prueba que se derivan de la especificación de los datos reales para ser procesados por el sistema.

El proceso de pruebas

IES INSTITUTO DE ESTUDIOS SUPERIORES



Etapas del proceso de pruebas

•Prueba de componentes o unidades

-Cada uno de los componentes son probados por separado;

-Componentes pueden ser funciones u objetos o agrupaciones coherentes de estas entidades.

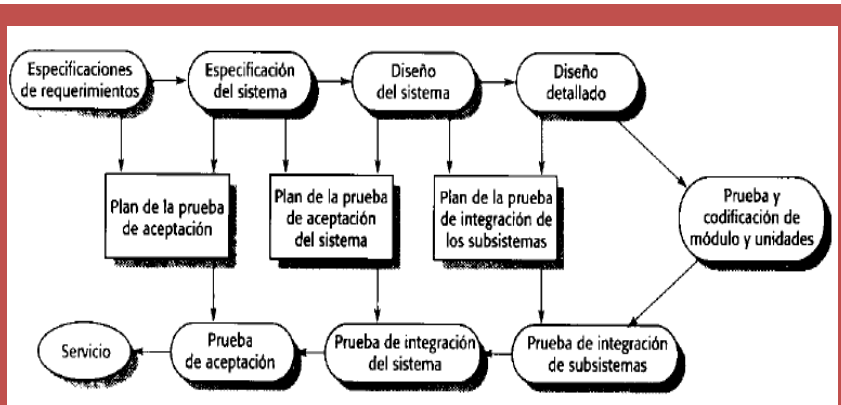
•Pruebas del sistema

-Prueba del sistema en su conjunto. Prueba de propiedades emergentes es particularmente importante.

•Pruebas de aceptación

-Pruebas con los datos de los clientes para comprobar que el sistema cumple con los requerimientos del cliente.

Fases de prueba



Actividades del proceso

- Especificación de software
- Diseño e implementación de software
- Validación de software
- **Evolución del software**

Evolución del software

- El software es inherentemente flexible y puede cambiar.
- Como los requerimientos cambian a través del cambio de las circustancias empresariales, el software que soporta la empresa también debe evolucionar y cambiar.

Evolución del sistema