

TÉCNICAS INTELIGENTES PARA PROCESOS DE DESARROLLO

CONTENIDO

1. UNIDAD 1. Conceptos Básicos
 - 1.1. Proceso de desarrollo de software
 - 1.2. Fases del Proceso de desarrollo de software
 - 1.3. Guías, Modelos y Estándares de la Ingeniería de Software
 - 1.4. Concepto de producto de software vs. Sistemas de información
 - 1.5. Problemática de los productos y procesos de software
 - 1.6. Evolución de la Ingeniería de Software

1 UNIDAD 1. CONCEPTOS BÁSICOS.

1.1 PROCESO DE DESARROLLO DE SOFTWARE

Un proceso Es el conjunto de actividades que interactúan con determinados elementos de entrada y que dan como resultado un producto o un servicio. cumple con ciertos requisitos establecidos, es una guía para todos los participantes: clientes, usuarios, desarrolladores y directivos ejecutivos. (Nos dice cómo interactúa con los clientes, etc.). Por lo tanto, un proceso es todo lo que se hace en una organización que tiene una entrada, hace una actividad con la entrada y crea una salida.

Nombre del Sistema o Proceso		
Entradas	Actividades	Salida
N	N	1 a N

✓ *Ejemplo 1: Proceso de elaboración de calzado artesanal.*

Entradas.

- *Material: refuerzos, hilos, plantilla, tacones, tijeras, gises, lápiz, pintura, tinta, cera, etiqueta de marca, plantilla para tallas, caja, bolsa.*
- *Lista de Proveedores*
- *Modelos del calzado*
- *Corte: piel, tela, forro, etc.*
- *Horma*
- *Mesa de corte*

Proceso.

- *Actividad 1.- Selección y adquisición del material.*

En este proceso se buscará el material idóneo (refuerzos, hilos, plantilla, tacones, tijeras, gises, lápiz, pintura, tinta, cera, etiqueta de marca, plantilla para tallas, caja, bolsa.) y se seleccionará el proveedor más adecuado.

- *Actividad 2.- Selección del modelo y corte a realizar.*

Se selecciona el modelo a fabricar, con el mejor corte y distribución más adecuada del corte, se marca y corta.

- *Actividad 3.- Armado de la Pala.*

Se unen el forro, refuerzos y corte para formar una sola pieza, ("Pala" es el nombre del corte cosido).

- *Actividad 4.- Montado.*

En este proceso es cuando se fija la pala en la horma, se monta en la palmilla y se fija con la entresuela.

- *Actividad 5.- Desmontado.*

En este proceso se retira la horma del zapato y se cosen la suela y la entresuela.

- **Actividad 6.- Terminado.**

Aquí se coloca el tacón, se rebajan los sobrantes, se introduce la plantilla, se tiñen, pulen y enceran los zapatos.

- **Actividad 7.- Empaquetado y distribución.**

Se pone la marca del zapato, se marca la talla, se embalan y se envían a la sección de venta o distribución.

Salidas.

Zapato artesanal ya embalado.

Nota. Embalar: acomodar, colocar convenientemente dentro de cubiertas los objetos que han de transportarse.

Nombre del Sistema o Proceso: Elaboración de calzado artesanal		
Entradas	Actividades	Salida
Material: refuerzos, hilos, plantilla, tacones, tijeras, gises, lápiz, pintura, tinta, cera, etiqueta de marca, plantilla para tallas, caja, bolsa. Lista de Proveedores Modelos del calzado Corte: piel, tela, forro, etc. Horma Mesa de corte	<i>Selección y adquisición del material.</i> <i>Selección del modelo y corte a realizar.</i> <i>Armado de la Pala.</i> <i>Montado.</i> <i>Desmontado.</i> <i>Terminado.</i> <i>Empaquetado y distribución</i>	<i>Zapato artesanal ya embalado</i>

Fuente:

https://www.ejemplode.com/58-administracion/2438-ejemplo_de_proceso.html#ixzz6CU0AS78V

Ejemplo 2:

✓ *Ejemplo del Proceso de Nóminas para empresas con pocas prestaciones.*

Nombre del Sistema o Proceso: Nómina		
Entradas	Actividades	Salida
Catálogo de trabajadores	Calcular el IMSS	Cuánto fue de seguro por trabajador
Número de días trabajados en el periodo		
Salario diario integrado (salario con aguinaldo y prima vacacional y vacaciones)		Lo que calculaste y retuviste a cada trabajador con lo del seguro
Número del seguro social de cada empleado		
Salarios mínimos de la zona A y B	Calcular el ISR	El resultado del cálculo del impuesto para el SAT
Tablas del ISR para cálculos de profesionales (Salarios mínimos)		
Salario base gravable (reparto de utilidades)		
Total de días trabajados	Calcular el sueldo	Cheques con los totales a pagar o archivo electrónico que se turnaría a una institución bancaria para que realice los depósitos
Salario de cada trabajador		Las afectaciones de las percepciones y deducciones aplicadas a las cuentas contables.
Incidencias: faltas, horas extras, retardos.		

Procedimiento: Es un documento en donde está establecido el proceso. (Esto es lo que indica ISO). El manual de procedimientos es un componente del sistema de control interno, el cual se crea para obtener una información detallada, ordenada, sistemática e integral que contiene todas las instrucciones, responsabilidades e información sobre políticas, funciones, sistemas y los procesos que se llevan a cabo en las distintas operaciones o actividades que se realizan en una organización. No todos los procesos deben tener forzosamente un procedimiento.

Tarea.

Realizar un proceso libre o alguno de los siguientes, por Equipos de 5 personas.

- Control de Inventarios.
- Farmacia

Nota:

Incluir portada.

Bibliografía en formato APA (En caso de haber utilizado alguna).

Cada alumno deberá subir el trabajo en aula virtual por favor.

Software:

De acuerdo con la Real Academia Española, en su Diccionario de la lengua española Vigésima segunda edición, la definición de software es la siguiente.

Software: (Voz inglesa). 1. m. *Inform.* Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

Esta definición no comprende todos los aspectos del software, debido a esto, a continuación, se presentan algunas definiciones más completas.

1. “El software no sólo son programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta.” (Sommerville, 2005)
2. “El software de computadora es el producto que los ingenieros de software construyen y después mantienen en el largo plazo. Incluye los programas de una computadora de cualquier tamaño y arquitectura, el contenido que se presenta conforme los programas se ejecutan y los documentos, tanto físicos como virtuales, que engloban todas las formas de medios electrónicos.” (Pressman, 2006)
3. “Programas o elementos lógicos que hacen funcionar a una computadora o una red, o que se ejecutan en ellos, en contraposición con los componentes físicos de la computadora o la red, agrupados bajo el nombre de hardware.” (Secretaría de Comunicaciones y Transportes, 2009)

Sommerville, 2005 menciona que un sistema de software consiste de: (**¡Error! No se encuentra el origen de la referencia.**):

- 1) diversos programas independientes,
- 2) archivos de configuración que se utilizan para ejecutar estos programas,
- 3) un sistema de documentación que describe la estructura del sistema,
- 4) la documentación para el usuario que explica cómo utilizar el sistema y,
- 5) en cuanto a los productos de software, sitios Web que permitan a los usuarios descargar la información de productos recientes.

Ilustración 1. Componentes de un sistema de software

En resumen, la mayoría de la literatura considerará que el **software** posee programas (conjuntos de instrucciones a través de las cuales se realizan ciertas tareas), estructuras de datos, que facilitan a las instrucciones manipular adecuadamente la información, documentación que describen el desarrollo, uso, instalación y mantenimiento de los programas y una configuración para el funcionamiento de los programas.

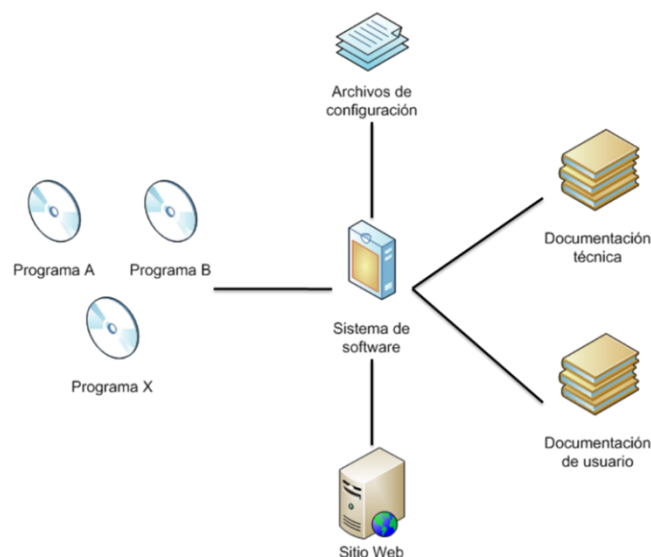
El software posee muchas más características que las ya mencionadas. Estas características lo diferencian de otros productos, como los productos manufacturados, que son productos hechos a mano o con auxilio de una máquina. La manufactura (del latín *manus*, mano, y *factura*, hechura) implica un proceso de fabricación industrial en la cual las materias primas son transformadas en bienes terminados a gran escala (Wikipedia, 2009).

Las **características** que diferencian al **software** son las siguientes (Pressman, 2006).

1. El software se desarrolla; no se manufactura.
2. El software no se desgasta; pero sí se deteriora.
3. La mayor parte del software aún se construye a la medida del cliente.

El software se desarrolla; no se manufactura, ya que el software se desarrolla a partir de un proceso de diseño intelectual y no a través de un proceso manual. El proceso de manufactura implica el paso de los materiales a través de distintas etapas o estaciones de trabajo, en las que se puede observar el proceso de construcción del producto muy claramente, ya que el producto está visible y es tangible.

En este contexto, la manufactura presenta un proceso más fácil de seguir que un desarrollo intelectual. Por ejemplo, siguiendo el proceso mostrado en la Ilustración 2 para la fabricación de una bicicleta, la primera etapa implica la construcción de la bicicleta (armar el cuadro, instalar cambios, pedales, cadenas, arandelas, asientos, salpicaderas, frenos, masas, piñones y tornillería), en la segunda etapa la bicicleta se somete a una prueba para constatar que las piezas están correctamente armadas y cada



parte cumple con su función, y finalmente el empaquetado incluye la preparación de la bicicleta para su transportación.

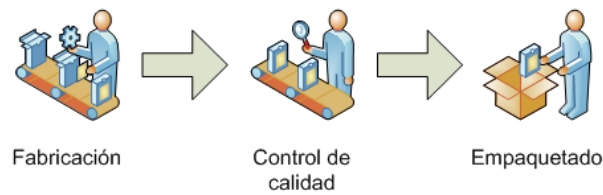


Ilustración 2. Proceso de fabricación de un producto manufacturado (Elaborado con Microsoft Visio 2007).

Un proceso de producción para una bicicleta se presenta en la Ilustración 3 (Instituto Nacional de Tecnología Industrial). Este proceso indica los pasos detallados a seguir para construir una bicicleta de manera clara y detallada.

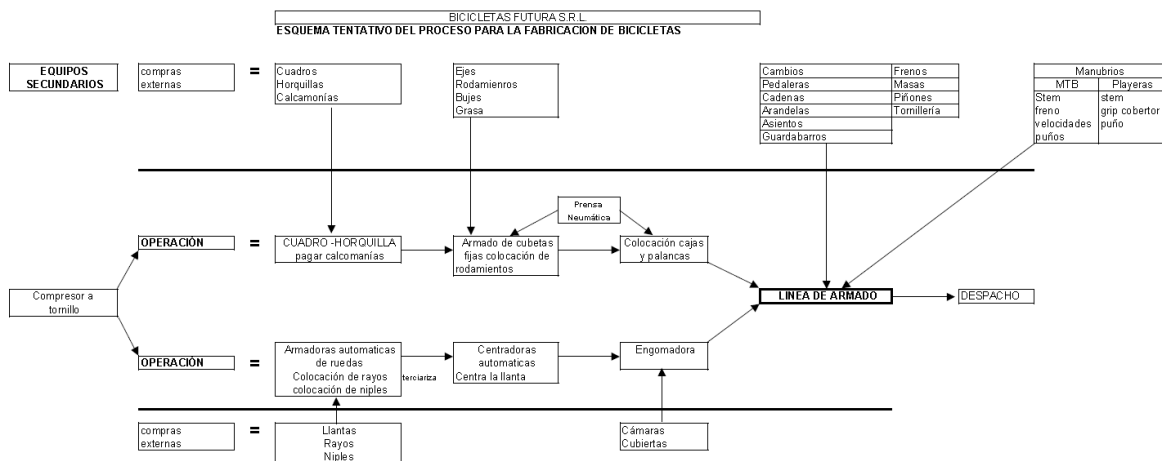


Ilustración 3. Proceso de producción de una bicicleta (Instituto Nacional de Tecnología Industrial)

En contraste se tiene el desarrollo de software (Ilustración 4) el cual se lleva a cabo como un proceso interno. Este tipo de desarrollo es particularmente difícil de seguir, ya que los productos se están desarrollando a través de un *proceso intelectual*. (proceso de desarrollo intelectual).



Ilustración 4. Proceso de desarrollo de software (Elaborado con Microsoft Visio 2007)

Una vez que el software ha sido desarrollado tiene un periodo de vida pero *no se desgasta; se deteriora*. Debido a que el software es un producto “intangible” no se ve sujeto a los efectos que sufre un producto “tangible” o manufacturado. En la Ilustración 5 se muestra la curva de la tasa de fallas para el software.

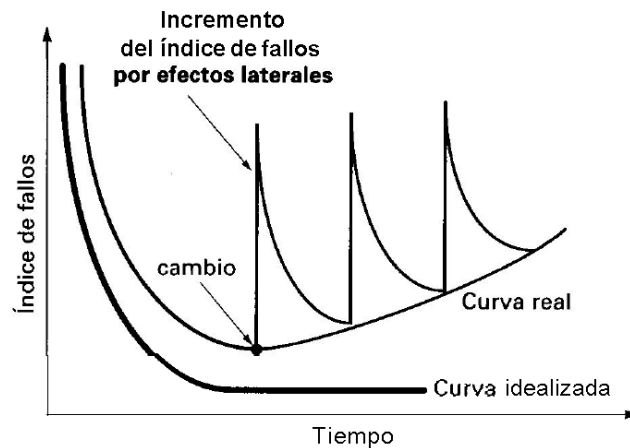


Ilustración 5. Curva de falla para el software (Pressman, 2006)

El deterioro del software es un síntoma de los defectos no corregidos durante su desarrollo, lo cual implica un aumento en el índice de fallos. Al corregir estos defectos, el software disminuye su índice de defectos. Más, sin embargo, el software sufre cambios constantemente, lo que implica que se pueden llegar a introducir más defectos y la tasa de fallos aumenta nuevamente, lo que representa un pico en la gráfica. En la Ilustración 5 se muestra la curva real del índice de defectos a través del tiempo de un software y su curva idealizada.

Resumiendo, las Características del Software:

- Elemento lógico, no físico.
- Desarrollado, no 'fabricado'.
- No se 'estropea', ¡se deteriora! deterioro por 'cambios')
- Mayoritariamente cerrado:
usar todo o nada
(poco ensamblaje de componentes, reutilización)

Clasificaciones del Software.

Existen diversas clasificaciones para los tipos de software existentes. Una de estas clasificaciones se basa en la forma en que el software es producido y al tipo de mercado al que va dirigido (Sommerville, 2005), clasificándolo en *software empaquetado* y en *software a la medida*.

- **Software empaquetado.** Son sistemas aislados desarrollados por organizaciones que se dedican al desarrollo y comercialización de productos al mercado abierto para el público en general. Normalmente este tipo de software cubren necesidades generales. El nombre de software empaquetado se debe a que su código fuente está "empaquetado", es decir, no puede ser visto o conocido por los usuarios o clientes que lo compran.
- **Software a la medida.** Estos son sistemas desarrollados por empresas que se dedican a producir software para clientes particulares. La característica principal de este software es que cubre en su totalidad las necesidades del cliente al que va dirigido.

Otra clasificación categoriza al software con base a su uso (Pressman, 2006):

- 1) Software de sistemas
- 2) Software de aplicación
- 3) Software científico y de ingeniería
- 4) Software de línea de productos
- 5) Aplicaciones basadas en Web
- 6) Software de inteligencia artificial

Existen más clasificaciones que implican diferentes usos del software en cuanto a su licenciamiento, así, bajo este contexto el software puede ser:

- freeware o software libre, cuya principal característica es que su código es abierto y distribuido con ciertas restricciones;
- shareware o software de prueba, que es distribuido para probar productos por un cierto periodo de tiempo para su posterior compra;
- software empaquetado.

Por lo tanto, un **proceso de desarrollo de software** es un conjunto de actividades y resultados asociados (código, procedimientos, reglas y documentación involucrada), que producen un producto de software a tiempo (Pressman, 2006).

El proceso del desarrollo de software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos de trabajo, se establecen los fundamentos, se asegura la calidad, y el cambio se maneja de manera apropiada (Sommerville, 2005).

Referencias:

Citado APA:

(A. 2012,10. Ejemplo de Proceso. Revista Ejemplode.com. Obtenido 10, 2012, de https://www.ejemplode.com/58-administracion/2438-ejemplo_de_proceso.html)
https://www.ejemplode.com/58-administracion/2438-ejemplo_de_proceso.html

Gómez Giovanni. (2001, diciembre 11). Manuales de procedimientos y su uso en control interno. Recuperado de <https://www.gestiopolis.com/manuales-procedimientos-uso-control-interno/>
<https://www.gestiopolis.com/manuales-procedimientos-uso-control-interno/>

Nueva ISO 9001:2015 - Escuela Europea de Excelencia:
<https://www.nueva-iso-9001-2015.com/2016/01/iso-9001-2015-diferencia-proceso-procedimiento/>

1.2 FASES DEL PROCESO DE DESARROLLO DE SOFTWARE

Existen 5 actividades fundamentales comunes para todos los procesos de desarrollo de software:

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.
5. **Documentación del software** que describe las especificaciones, el desarrollo, uso, instalación y mantenimiento de los programas.

Distintos procesos de software organizan las actividades de diferentes formas y las describen con diferente nivel de detalle.

- El tiempo de cada actividad varía, así como los resultados.
- Organizaciones diferentes usan procesos diferentes para producir el mismo producto.
- Para diferentes tipos de software se emplean diferentes procesos de desarrollo: El software de una institución bancaria tiene que ser completamente especificado antes de que pueda comenzarse su desarrollo. En contraste un sitio Web informativo puede ser especificado y desarrollado de forma paralela. Por ende, las actividades genéricas pueden reorganizarse acorde a la necesidad del proyecto y del producto.

Es importante recalcar que un proceso de software inadecuado puede afectar la calidad del producto final reduciéndola, o incluso incrementar el costo del proyecto.

Ciclo de vida \neq Ciclo de desarrollo.

- De vida: Dura toda la vida del sistema. Desde la concepción hasta el fin de su uso.
- De desarrollo: Se inicia desde el análisis hasta la entrega del usuario.

1.3 GUÍAS, MODELOS Y ESTÁNDARES DE LA INGENIERÍA DE SOFTWARE

1.3.1 MODELO DE PROCESOS DE SOFTWARE

Para resolver los problemas reales de una industria, un ingeniero de software o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso, métodos, herramientas y fases genéricas. Esta estrategia a menudo se llama *modelo de proceso* para la Ingeniería de Software o *paradigma de Ingeniería de Software*. Se selecciona un modelo de proceso para la ingeniería de software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren. (Pressman, 2006)

La mayor parte de los modelos de procesos de software se basan en uno de los tres modelos generales o paradigmas de desarrollo de software: (Sommerville, 2005)

1. *El enfoque en cascada*. Considera cada actividad y las representa como fases de procesos separados, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etc. Después de que cada etapa queda definida – se afirma – y el desarrollo continúa con la siguiente etapa.
2. *Desarrollo iterativo*. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste 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 re-implementar utilizando un enfoque más estructurado para producir un sistema más sólido y mantenible.
3. *Ingeniería de 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.

Con independencia del modelo de proceso que se seleccione para un proyecto de software, todas las etapas coexisten simultáneamente en algún nivel de detalle o refinamiento. (Pressman, 2006).

1.3.2 GUIAS Y ESTANDARES

Guía: conjunto de criterios bien definidos y documentados que encaminan una actividad o tarea. Es más flexible que un estándar.

Estándar: conjunto de criterios aprobados, documentados y disponibles para determinar la adecuación de una acción (*estándar de proceso*) o de un objeto (*estándar de producto*).

Ventajas

- Agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software.
- Engloban los “conocimientos” que son patrimonio de una organización.
- Proporcionan un marco para implementar procedimientos de aseguramiento de la calidad.
- Proporcionan continuidad entre el trabajo de distintas personas.

Tipos de estándares

- **Estándares para datos:**
Desde asignar nombres a los datos y especificar longitud y tipo hasta los relacionados con BD p.ej., SQL1999, ODMG 2.0
- **Estándares de codificación:**
abreviaturas y designaciones formales para describir actividades dentro de la organización
- **Estándares estructurales:**
políticas de división del software en módulos
- **Estándares de documentación** *Draft Standard for Software User Documentation*, IEEE Std1063/D5.1. 2001
- **Estándares de proceso software** modelo de madurez de capacidades CMM
- **Estándares para otras actividades**

Tipos de estándares para el producto de software

IEEE Standards Collection Software Engineering – 1998 Edition
IEEE Std. 829-1983, Standard for Software Test Documentation
IEEE Std. 830-1993, Recommended Practice for Software Requirements Specifications.
IEEE Std. 1045-1992, Standard for Software Productivity Metrics
IEEE Std. 1062-1987, Recommended Practice for Software Acquisition
IEEE Std. 1063- 1987, Standard for Software User Documentation
IEEE Std. 1219-1992, Standard for Software Maintenance

Otros Estándares relacionados con el proceso de software.

Familia ISO 9000 ⇒ CALIDAD

IEEE 1074-1998 - IEEE Standard for Developing Software Life Cycle Processes

IEEE – Institute of Electrical and Electronics Engineers

ISO – International Organization for Standardization

IEC – International Electrotechnical Commission

1.4 CONCEPTO DE PRODUCTO DE SOFTWARE VS. SISTEMAS DE INFORMACIÓN

1.4.1 PRODUCTO DE SOFTWARE

Es una entidad diseñada para ser entregada a un usuario final. [ISO/IEC 9126: 1991]

1.4.2 SISTEMA DE INFORMACIÓN

Es un sistema conformado por: software, hardware, procedimientos, datos y operadores insertados en un proceso organizacional. [Adaptado de Mora et al, 2002]

1.5 PROBLEMÁTICA DE LOS PRODUCTOS Y PROCESOS DE SOFTWARE

La ingeniería de software cubre todos los aspectos del desarrollo de software y en el siglo XXI, 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.*
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 ya que ese es el objetivo de la Ingeniería de Software. 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.*
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, por eso se dice que la ingeniería de software es considerada como una disciplina.

A continuación vamos a conocer los problemas generados por productos de software de mala calidad, así como los problemas de los procesos respecto a cumplimiento de presupuestos, especificaciones y calendarios.

Proyectos	1994	1996	1998	2012
Exitosos	16%	27%	26%	39%
Presupuesto Excedido	53%	33%	46%	43%
Cancelados	31%	40%	28%	18%

1.5.1 ALGUNAS CIFRAS

Monto Asignado	Éxito	Personal	Tiempo(meses)
< 750,000	55%	6	6
> 750,000 y < 1.5M	33%	12	9
1.5M a 3M	25%	25	12
3M a 6M	15%	40	18
6M a 10M	8%	+250	+24
> 10M	0%	+500	+36

Números de los Proyectos de Software

Usado con trabajo extra o abandonado después	19%
Pagado, pero nunca entregado	29%
Entregado, pero nunca usado	47%
Usado después de cambios	3%
Usado tal como se entregó	2%

Un estudio del Standish Group reveló que en un sistema típico las funcionalidades que:

"se usan siempre" son el	7%,
"se usan a menudo" el	13%,
"algunas veces" el	16%,
"raras veces" el	19%
y "nunca" el	45%.

Problemas Actuales del desarrollo de Software

- Incapacidad para estimar tiempo, costo y esfuerzo para el desarrollo de un producto software.
- resulta difícil constatar el progreso del proyecto de sw
- Se extiende el tiempo de entrega y presupuestos
- Falta de calidad del producto software.
- No podemos encontrar todos los errores antes de entregar el software a nuestros clientes
- Falta de metodologías de trabajo en equipo

Causas	Efectos
<i>Poco esfuerzo en el análisis y el diseño.</i>	<i>Se gastó mucho tiempo en pruebas. El costo de reparación fue mayor que el costo de desarrollo. La mayoría de los cambios son consecuencia de nuevos requerimientos.</i>
No hay capacitación en metodologías de desarrollo (planeación y calidad)	Problemas de comunicación con los clientes y con compañeros de equipo (no hay proceso definido, no hay equipo) El trabajo está dirigido en seguir las actividades fijadas y no en la calidad.
El directivo compromete a metas inalcanzables	El sistema estará en 5 meses

1.5.2 ALGUNOS CASOS EJEMPLARES

- LAS (London Ambulance Service) 1990-1992
 - Hubo pérdidas humanas
 - Fallas en el proceso de desarrollo
- TAURUS (London Exchange Stock) 1986-1992
 - 600 millones de libras en pérdidas
 - Exceso de confianza en la TI misma
 - No consideraciones sociopolíticas
- CONFIRM System (Traveling Reservation) 1988-1990
 - 125 millones de USD en pérdidas
 - Subestimación de costos y calendario
 - Fallas en gestión del proyecto

1.5.3 PROBLEMAS PRINCIPALES

- El producto de software no cumple las especificaciones acordadas
- El producto de software fue desarrollado a un costo mayor que el planeado
- El producto de software fue desarrollado en un tiempo mayor al acordado

1.5.4 CAUSAS DE LOS PROBLEMAS

- El Desarrollo de Software es un proceso complejo
- El Desarrollo de Software realizado sin control administrativo generalmente conduce a fallas críticas
- El Desarrollo de Software requiere utilizar procesos administrativos y técnicos sólidos.

1.6 EVOLUCIÓN DE LA INGENIERÍA DE SOFTWARE

En 1968 en una conferencia de NATO (OTAN- Organización del Atlántico Norte), se concibe el concepto de “Ingeniería de Software”.

70’S A 80’S

Se proponen diversos **Ciclos de Vida de Desarrollo de Sistemas** (Systems Development Life Cycle), los cuales definen las etapas y la secuencia a seguir entre ellas para desarrollar un producto de software. Ejemplos:

- Ciclo de Cascada [Bohem, 1981]
- Ciclo de Prototipos [Brooks, 1975; Gomma & Scott, 1981]
- Ciclo Espiral [Bohem, 1988]

90’S – PRESENTE

Se proponen **Estándares de Procesos** para ser usados en **Ciclos de vida de desarrollo de Sistemas**, los cuales definen no sólo etapas sino también los **procesos** a seguir para desarrollar un producto de software. Ejemplos:

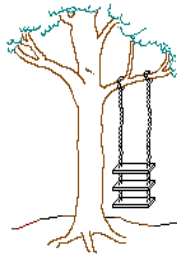
- ISO-12207 (1995)
- ISO-15288 (2000)
- RUP (Rational, 2000)

Se proponen **Modelos de Madurez y/o de Mejoramiento de Procesos** para evaluar el grado en que los **procesos** usados en los **Ciclos de vida de Desarrollo de Sistemas** son explícitamente definidos, administrados, medidos, controlados y efectivos. Ejemplos:

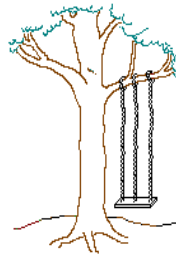
- CMM-SW (1995)
- CMMI-SW (2000)
- ISO-15504 (2000) (Conocido como SPICE)

Sensibilizar a desarrolladores, usuarios y a la organización misma que los productos de software requieren de un **proceso** al igual que otros productos complejos para ser desarrollados.

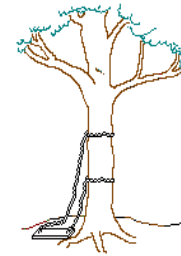
Resumiendo, veamos el siguiente ejemplo:



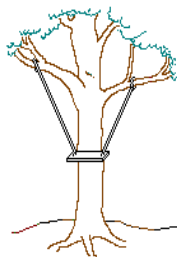
1. Lo que el director desea.



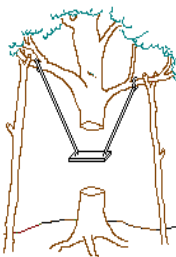
2. Como lo define el director de proyecto.



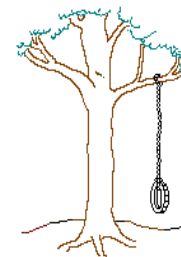
3. Como se diseña el Sistema.



4. Como lo desarrolla el programador.



5. Como se ha realizado la instalación.



6. Lo que el usuario quería.