

DIEGO GUERRERO PEÑA

Elementos básicos de ingeniería del software

**ELEMENTOS BÁSICOS
DE INGENIERÍA DEL SOFTWARE**

DIEGO GUERRERO PEÑA





INSTITUTO TECNOLÓGICO METROPOLITANO
Institución Universitaria

ELEMENTOS BÁSICOS DE INGENIERÍA DEL SOFTWARE

© Diego Guerrero Peña

© Instituto Tecnológico Metropolitano

1a. Edición: septiembre de 2007

ISBN: 978-958-98275-6-7

Dirección editorial
Fondo Editorial ITM

Diagramación y montaje
L. Vieco e Hijas Ltda.

Impreso y hecho en Medellín, Colombia

Las opiniones, originalidad y citas del texto son de la responsabilidad del autor. El Instituto salva cualquier obligación derivada del libro que se publica. Por lo tanto, ella recaerá única y exclusivamente en el autor.

Instituto Tecnológico Metropolitano
Calle 73 No. 76A 354
Tel.: (574) 440 51 00
Fax: 440 51 01
www.itm.edu.co
Medellín - Colombia

Agradezco a los profesores Carlos Builes y Alejandro Villa, a los estudiantes de la asignatura Ingeniería del Software del ITM, por su colaboración y aportes al libro. Igualmente, a Roger Pressman, autor del libro *Ingeniería del software: un enfoque práctico*; a Craig Larman, autor del libro *UML y patrones introducción al análisis y diseño orientado a objetos*; y a aquellos autores que aparecen en la bibliografía de mi texto, quienes han logrado propuestas de evolución del desarrollo de software.

Guardo gratitud, también, con mis seres queridos que tuvieron la paciencia y tolerancia de mi ausencia por dedicarme al desarrollo del texto.

Sin la confianza en mis conocimientos, y en mí como profesional, no hubiera podido dedicar tiempo a esta larga labor; por ello, doy especiales agradecimientos a mi jefe y amigo Fray León Osorio, decano de Sistemas de Información del ITM.

TABLA DE CONTENIDO

PREFACIO	17
INTRODUCCIÓN	19
1. CONCEPTUALIZACIÓN BÁSICA.....	23
1.1 Sistema	23
1.2 La organización.....	23
1.3 Enfoque de sistemas	24
1.4 Sistemas de información	25
1.4.1 Definiciones	25
1.4.2 Elementos.....	27
1.4.3 Papel de los sistemas de información en las organizaciones.....	28
1.4.4 Administración de sistemas de información	30
1.5 Consideraciones de software	32
1.5.1 Definición.....	32
1.5.2 Evolución.....	32
1.5.3 Crisis del software.....	33
1.5.4 Factores críticos de éxito en el desarrollo de sistemas de información.....	34
1.6 Ciclo de vida de un sistema de información (SIBTIC).....	34
1.6.1 Preanálisis o anteproyecto	35
1.6.2 Análisis.....	35
1.6.3 Diseño.....	35
1.6.4 Construcción	35
1.6.5 Pruebas	36
1.6.6 Implementación. (puesta en marcha del sistema)	36
2. EL PROCESO DE DESARROLLO DEL SOFTWARE.....	37
2.1 La ingeniería del software: una tecnología estratificada	38
2.2 El proceso del software	40
2.2.1 CMM un modelo para la mejora de procesos de software	41
2.2.2 Modelos de desarrollo del software	42
2.2.2.1 Modelo lineal secuencial	45

2.2.2.2	Modelo en cascada.....	47
2.2.2.3	Metodología por fases.....	48
2.2.2.4	Modelo de construcción de prototipos	48
2.2.2.5	Modelo desarrollo rápido de aplicaciones.....	51
2.2.2.6	Modelos evolutivos en proceso del software.....	53
2.3	Fases o etapas de un proyecto informático	66
2.3.1	Planeación (preanálisis o anteproyecto)	67
2.3.2	Ciclo de vida de desarrollo (ejecución real).....	68
2.3.3	ciclo de vida de operación (entrega y ejecución por parte de los usuarios).....	68
3.	ANTEPROYECTO O PREANÁLISIS	69
3.1	Consideraciones generales.....	69
3.1.1	Objetivos.....	69
3.1.2	Definición.....	69
3.2	Importancia del estudio de factibilidad.....	70
3.3	Características del estudio de factibilidad	71
3.4	Talento humano necesarios.....	71
3.5	Pasos en el desarrollo del estudio de factibilidad	72
3.5.1	Reconocimiento general del sistema.....	72
3.5.1.1	Ubicación general.	72
3.5.1.2	Delimitación o alcance del sistema.....	73
3.5.1.3	Objetivos del sistema.	73
3.5.1.4	Definición del sistema	75
3.5.2	Recursos para el desarrollo del nuevo sistema	76
3.5.2.1	Recursos económicos.	76
3.5.2.2	Recursos de personal.	76
3.5.2.3	Recursos técnicos.....	76
3.5.2.3.1	Hardware o equipo disponible	77
3.5.2.3.2	Software.....	77
3.5.2.3.2.1	De soporte	77
3.5.2.3.2.2	De utilidad.....	77
3.5.3	Estimativos de desarrollo del sistema.....	77
3.5.4	Análisis de factibilidad del sistema	78
3.5.4.1	Factibilidad financiera.....	78
3.5.4.2	Factibilidad técnica.....	82
3.5.4.3	Factibilidad operativa	82

3.5.5	Alternativas y recomendaciones	83
3.5.6	Cronograma de actividades	84
3.5.7	Decisión final	88
3.6	Resumen	88
4	ANÁLISIS.....	91
4.1	Consideraciones generales.....	91
4.1.1	Objetivos.....	92
4.1.1.1	Especificación informal del sistema de clasificación de paquetes	93
4.1.1.2	Función del sistema.....	93
4.1.1.3	Información que se maneja	93
4.1.1.4	Interfaces del sistema.....	94
4.1.1.5	Rendimiento.....	94
4.1.1.6	Restricciones.....	94
4.1.1.7	Asignación de funciones.....	95
4.1.2	Definición.....	97
4.2	importancia del análisis.....	98
4.3	Características del análisis	99
4.4	Participación requerida.....	99
4.5	Pasos en el desarrollo del análisis	100
4.5.1	Análisis del sistema actual.....	101
4.5.2	Análisis requerimientos del nuevo sistema	102
4.5.3	Revaluación del preanálisis.....	102
4.5.4	Desarrollo de las especificaciones del sistema propuesto	102
4.5.5	Revisión del análisis.....	103
4.5.6	Entrega a la etapa de diseño	104
4.6	Metodologías de análisis	104
4.6.1	Metodologías tradicionales	104
4.6.2	Metodologías estructuradas.....	105
4.7	Análisis de requisitos	106
4.7.1	Vistazo rápido	107
4.7.2	Identificación de requisitos para el software	110
4.7.2.1	Inicio del proceso	111
4.7.2.2	Técnicas para facilitar las especificaciones de una aplicación.....	112

4.7.3	Principios de especificación	118
4.7.3.1	Es necesario separar funcionalidad e implementación.....	118
4.7.3.2	El lenguaje de especificación debe estar orientado al proceso.	118
4.7.3.3	La especificación debe abarcar todo el sistema del que el software es parte.	119
4.7.3.4	La especificación debe abarcar también el entorno del sistema.....	119
4.7.3.5	La especificación debe modelar el dominio del problema.....	119
4.7.3.6	La especificación debe ser operativa.	119
4.7.3.7	La especificación debe ser ampliable y tolerante a la incompletitud.....	119
4.7.3.8	La especificación debe estar localizada y débilmente acoplada.	119
4.7.3.9	Es necesario separar funcionalidad e implementación.....	119
4.7.3.10	El lenguaje de especificación debe estar orientado al proceso.	119
4.7.3.11	La especificación debe abarcar todo el sistema del que el software es parte.	120
4.7.3.12	La especificación debe abarcar también el entorno del sistema.....	120
4.7.3.13	La especificación debe modelar el dominio del problema.....	120
4.7.3.14	La especificación debe ser operativa.	121
4.7.3.15	La especificación debe ser ampliable y tolerante a la incompletitud.....	121
4.7.3.16	La especificación debe estar localizada y débilmente acoplada.	122
4.7.3.17	Especificación del sistema	122
4.7.3.18	Evaluación inicial de la especificación	123
4.7.3.19	Se ha delimitado correctamente el ámbito del proyecto.....	123
4.7.3.20	Se ha definido correctamente la funcionalidad, las interfaces y el rendimiento.....	123
4.7.3.21	Las necesidades del usuario y el análisis de riesgos justifican el desarrollo del proyecto.	123

4.7.3.22	Cliente y analista tienen la misma percepción de los objetivos del proyecto.	123
4.7.3.23	Evaluación técnica de la especificación	123
4.7.3.24	Las estimaciones de riesgos, coste y agenda se corresponden con la complejidad del proyecto.	123
4.7.3.25	Todos los detalles técnicos (asignación de funciones, interfaces, rendimientos) están bien definidos.	123
4.7.3.26	La especificación del sistema sirve de base para las fases siguientes (en concreto para la ingeniería del software)...	123
4.7.3.27	Especificación del software	124
5	ANÁLISIS ESTRUCTURADO	129
5.1	Conceptos generales	129
5.1.1.	Definición	129
5.1.2	Características	129
5.1.3	Componentes	129
5.2	Modelo de procesos	130
5.2.1	Propósito	131
5.2.2	Diagramas de flujos de datos (DFD)	131
5.2.2.1	Flujos de datos	131
5.2.2.2	Procesos o transformaciones	132
5.2.2.3	Almacenamientos	132
5.2.2.4	Son los lugares donde se acumula la información dentro del sistema, con el fin de ser utilizada por procesos posteriores	132
5.2.2.5	Terminales, entidades o agentes externos	133
5.2.2.6	Como construir un diagrama de flujo de datos	133
5.2.2.7	Nivelación	134
5.2.2.7.1	Diagrama de contexto (nivel 0)	135
5.2.2.7.2	Diagrama de flujo de datos general (nivel 1)	135
5.2.2.7.3	Diagrama de flujo de datos de nivel 2	135
5.2.2.7.4	Diagrama de flujo de datos de nivel 3 al nivel n	135
5.2.2.8	Balanceo	136
5.2.2.9	Convenciones numéricas	137
5.2.2.10	Otros aspectos	137
5.2.3	Diccionario de datos (DD)	138
5.2.3.1	Componentes del diccionario de datos	138

5.2.3.1.1	Unidades externas o entidades o agentes externos	138
5.2.3.1.2	Procesos	139
5.2.3.1.3	Almacenamientos	139
5.2.3.1.4	Flujo de datos	140
5.2.3.2	Componente de datos	141
5.2.3.3	Elemento de datos	141
5.2.3.4	Convenciones para el diccionario	141
5.2.4	Eventos	142
5.2.4.1	Propósito	142
5.2.4.2	Definición	142
5.2.4.3	Listado de eventos	144
5.2.4.4	Diccionario de eventos	144
5.2.4.5	Matrices de eventos	147
5.2.4.5.1	Matriz evento/ubicación del negocio	147
5.2.4.5.2	Crud evento/entidad	147
5.2.4.5.3	Crud ubicación/entidad	148
5.2.4.6	Descubrimiento de eventos	148
5.2.4.7	Tipos de eventos	148
5.2.4.8	Revisión de tipos de eventos	149
5.2.4.9	Organización de la lista de eventos	150
5.2.4.10	Nivelación de eventos	151
5.3	Modelo de información	154
5.3.1	Propósito	154
5.3.2	Entidades	154
5.3.3	Relaciones	155
5.3.4	Cardinalidad	156
5.3.5	Atributos y sus relaciones	157
5.3.6	Tipos de entidades	161
5.3.6.1	Entidad atributiva	161
5.3.6.2	Entidad asociativa	162
5.3.7	Definición de atributos	164
5.3.8	Matrices de entidad	164
5.3.8.1	Crud evento/entidad	165
5.3.8.2	Matriz evento/ubicación del negocio	165
5.2.8.3	Crud ubicación/entidad	166
5.2.8.4	Matriz autorización usuario/entidad	166

5.3.9	Algunos ejemplos de modelos de información	167
5.4	Especificaciones de proceso (miniespecificaciones)	167
5.4.1	Aspectos a tener en cuenta.....	168
5.4.2	Formas de describir mini especificaciones.....	168
5.4.2.1	Español estructurado	168
5.4.2.2	Convenciones	169
5.5	Balaceo de los modelos.....	171
5.5.1	Balaceo del DFD y el EL DD.....	175
5.5.2	Balaceo del dfd y la especificación de proceso	175
5.1.1	Balaceo de las especificaciones del proceso con el DFD y el DD	176
5.5.4	Balaceo del diccionario de datos con el DFD y las especificaciones del proceso	178
5.5.5	Balaceo del DER con el DFD y las especificaciones de proceso	179
5.5.6	Balaceo del dfd y el diagrama de transición de estados	181
5.7	Resumen capítulos del análisis de requerimientos	182
6	DISEÑO	183
6.1	Consideraciones generales.....	183
6.1.1	Objetivos.....	183
6.1.2	Definición.....	183
6.2	Importancia del diseño	184
6.3	Características del diseño.....	185
6.4	Participación requerida	186
6.5	Pasos en el desarrollo del diseño	186
6.5.1	Definir la estructura del sistema (Diseño Global).....	187
6.5.1.1	Pasos para el desarrollo de la estructura	187
6.5.1.2	Descomposición funcional de módulos	188
6.5.2	Diseño detallado (Diseño del interior de cada módulo) ...	190
6.5.2.1	Requerimientos para el diseño detallado.....	190
6.5.2.2	Atributos de un módulo.....	191
6.5.3	Diseño de bases de datos	193
6.5.4	Diseño de entradas y salidas.....	193
6.5.4.1	Diseño de documentos fuentes	194
6.5.4.1	Diseño de ventanas	194

6.5.4.3	Diseño de la interacción	201
6.5.4.4	Diseño de reportes	209
6.5.5	Diseño de operación del sistema (prototipos)	210
6.5.6	Documentación y revisión	212
7.	DISEÑO ESTRUCTURADO	213
7.1	Características	213
7.2	Componentes	213
7.2.1	Identificación de módulos	213
7.2.1.1	Tipos de módulos	214
7.2.1.2	Interfaces entre módulos.....	214
7.2.2	Diagrama de estructura	215
7.3	características a evaluar en el diseño	216
7.3.1	Acoplamiento.....	216
7.3.2	Cohesión.....	218
7.3.3	Factorización.....	223
7.1.1	Fan In	224
7.3.5	Fan Out	224
7.4	Procedimiento para construir la estructura del sistema a través del diseño estructurado.....	225
7.5	Resumen de los capítulos de diseño	226
8.	BIBLIOGRAFÍA	229
	Referencias Adicionales:	230
	Término en inglés término equivalente en español	231
	Apéndice	234
9.	Anexos	237

PREFACIO

Elementos básicos de Ingeniería del Software no pretende ser una obra acabada. Al contrario, su intención es servir como elemento de discusión académica para el planteamiento de una guía práctica sobre los elementos básicos de la ingeniería del software, que ayude a los estudiantes a comprender las etapas de pre análisis, análisis y diseño del desarrollo de sistemas informáticos.

Esta primera edición recoge las experiencias propias del autor en desarrollo de SIBTIC, al igual que experiencias en la asesoría de proyectos de grado de programas informáticos en diferentes instituciones de educación superior, enfocándose principalmente en el análisis y diseño estructurado. En una futura edición se tratarán los temas de análisis y diseño con enfoque objetual.

Su referente temático se basa en libros, tales como: *Notas de Ingeniería* de Carlos Builes (sin editar 2003), *Ingeniería del software: un enfoque práctico*, de Roger Pressman (5ª edición), *UML y patrones* de Craig Larman (1º edición), trabajos elaborados por los estudiantes Juan Alberto Paddilla y Carlos Eduardo Rubiano Díaz, en la asignatura de Ingeniería del Software Cliente-Servidor, del programa académico Ingeniería de Sistemas de Información del Instituto Tecnológico Metropolitano, entre sus principales fuentes de información.

INTRODUCCIÓN

Elementos básicos de Ingeniería del Software recopila algunas de las teorías y técnicas para el desarrollo de sistemas de información basados en tecnologías de información computarizadas (SIBTIC).

El texto académico permite al estudiante tener una guía práctica para el desarrollo* de Sistemas de Información (SIBTIC), además de tener una referencia bibliográfica de consulta en todo momento. Se encuentra organizado de acuerdo con el ciclo de vida que se sigue en el desarrollo de un sistema de información, presentando en cada etapa de éste, sus características, objetivos y herramientas disponibles para su elaboración y verificación. Esta primera edición presenta los elementos principales del enfoque estructurado.

Elementos básicos de Ingeniería del Software contiene, en su primer capítulo, la conceptualización básica, presenta la terminología que el estudiante debe conocer para la comprensión de los temas posteriores; se explicitan los elementos organizacionales, el enfoque de sistemas y su terminología principal, los elementos que componen un sistema informático, las consideraciones generales del software con respecto a su definición, evolución, crisis por la que ha pasado; finaliza este capítulo con los factores críticos de éxito que se deben tener en cuenta en los procesos de desarrollo de sistemas informáticos.

El segundo capítulo contiene el proceso de desarrollo del software, el modelo CMM para la mejora de procesos de software y los modelos de desarrollo de software más utilizados; las fases de un proyecto informático: planeación, ciclo de vida de desarrollo y ciclo de vida de operación.

* Su alcance es la elaboración de aplicativos de software con fines únicamente académicos.

El tercer capítulo presenta de forma detallada la etapa de anteproyecto o pre análisis, contemplando los elementos generales, la importancia y características del estudio de factibilidad del proyecto, enfatiza la factibilidad financiera, es decir, la rentabilidad del proyecto.

Este capítulo también contiene los elementos fundamentales para tener en cuenta con respecto al talento humano que requiere un proyecto, y termina con los pasos en el desarrollo del estudio de factibilidad, elemento esencial para la decisión de ejecutar o no el proyecto (éste comienza con el reconocimiento general del sistema, actividad que pretende lograr una ubicación general dentro de la organización, tanto del área en estudio como del sistema en sí mismo, y proporciona una primera visión de lo que será el alcance del nuevo sistema).

Aquí se definen, también, los recursos técnicos, tecnológicos y humanos, dispuestos en la empresa para el desarrollo del sistema en cuestión; se definen los estimativos de los recursos que realmente se necesitan para llevar a buen punto el proyecto, y realizar el análisis de factibilidad (financiera, técnica, etcétera.). Una vez terminada la factibilidad del sistema, se realizan las posibles alternativas de solución, es decir, las distintas estrategias de cómo se puede implementar el sistema de información, para continuar con el tema del cronograma de actividades, el cual debe ser realizado en una hoja de cálculo y en un sistema manejador de proyectos. Luego se brindan los elementos para la decisión final con respecto a la realización o no del sistema informático. El capítulo termina con un resumen del mismo, empleando la técnica de entrada - proceso - participación - salida.

En el cuarto capítulo se dan los elementos sobre el análisis de requerimientos, comenzando con las consideraciones generales, luego definiendo los elementos que lo hacen importante, las características que lo componen y la participación de las personas dentro del proyecto. Se continúa con los pasos en el desarrollo de la etapa de análisis, definiendo cada una de sus actividades y tareas a desa-

rollar. A continuación, se presentan las metodologías tradicionales y las estructuradas; por último, se trabajan los elementos fundamentales del análisis de requisitos, sus fundamentos, sus características, sus actividades, etcétera.

En el quinto capítulo se trabaja sobre la metodología de análisis estructurado. Comienza con los conceptos generales, sigue con el modelo de procesos, donde se identifican todos los procesos de la organización para definir con cuáles trabajar el sistema de información por desarrollar, utilizando para ello las técnicas de diagrama de flujo de datos, diccionario de datos y eventos. Luego se estudian los elementos del modelo de información, en donde se trabaja con las bases de datos relacionadas y su fundamentación, entidades, relaciones entre ellas, cardinalidad, modalidad, atributos, para desarrollar las matrices de entidad, que verifican la coherencia del modelo de información elaborado. Termina la sección con algunos ejemplos de modelos de información. A continuación, se sigue con la especificación de los procesos que pertenecen al diagrama de flujo de datos, dando referentes para tener en cuenta en el modelado e indicando las formas de describir las mini especificaciones; es decir, los macro algoritmos que componen los diferentes procesos. Finaliza con el balanceo de los modelos vistos, donde se especifica cada uno de los modelos estudiados y cómo cada modelo se interrelaciona con los demás modelos, buscando un equilibrio en el modelado del sistema de información, ya que cada una de estas herramientas se enfoca en un aspecto crítico del sistema por modelar.

El capítulo termina con un resumen del capítulo anterior y de éste, para relacionar los elementos del análisis, empleando la técnica de entrada - proceso - participación - salida.

El sexto capítulo trata sobre la etapa de diseño. Comienza con las consideraciones generales, objetivos y definición; continúa con la importancia del diseño en el proceso de desarrollo, sus principales características, la participación requerida del recurso humano en sus actividades para especificar los pasos que se deben seguir en el desarrollo del diseño. Estos últimos incluyen la definición de

la estructura del sistema –diseño global, diseño detallado– diseño en el interior de cada módulo, diseño de bases de datos, diseño de entradas y salidas. En éste se especifica toda la interacción usuario- máquina, teniendo en cuenta el diseño de documentos fuentes, diseño de ventanas (pantallas), diseño de la interacción, posibles errores que se pueden cometer en el diseño y, por último, en el diseño de entradas y salidas, el diseño de reportes, para continuar con los siguientes pasos que son: diseño de operación del sistema (prototipos), la documentación y los pasos de la revisión.

En el séptimo capítulo se trabaja sobre la metodología de diseño estructurado. Comienza con las características principales, luego analiza cada uno de sus componentes, iniciando con la identificación de los módulos, donde se especifica tipos de módulos, interface entre ellos. Sigue con el diagrama de estructura, el cual muestra la organización relacional y, en muchos casos jerárquica, de los diferentes módulos que componen el sistema de información, además de las relaciones existentes entre ellos. Sigue con las características para tener en cuenta cuándo se evalúa el diseño, acoplamiento, cohesión, factorización, fan in, fan out; continúa con el procedimiento para construir la estructura del sistema a través del diseño estructurado. El capítulo termina con un resumen del capítulo anterior y de éste, para relacionar los elementos del diseño, empleando la técnica de entrada - proceso - participación - salida.

En el capítulo ocho se encuentra la bibliografía empleada para el desarrollo del libro, al igual que hipervínculos a páginas web y un glosario general de términos imprescindibles para la lectura del texto.

Por último, se adiciona el anexo 1, con un ejemplo de un sistema informático desarrollado por los estudiantes del ITM, Juan Alberto Padilla y Carlos Eduardo Rubiano Díaz, utilizando las técnicas expuestas en el texto. El anexo contiene el pre análisis, análisis y diseño, el cual sirve como base para que los alumnos desarrollen sus propios proyectos académicos.

1. CONCEPTUALIZACIÓN BÁSICA

1.1 SISTEMA

Es un conjunto de elementos que se interrelacionan para lograr un objetivo común. Ejemplos: la organización (empresa), el sistema circulatorio, la universidad, etcétera.

Un sistema posee características, tales como: límites, objetivos, medio ambiente, entradas (del medio al sistema), salidas (del sistema al medio), elementos, relaciones (interrelación entre elementos), funciones (transformación de entradas en salidas), retroalimentación.

1.2 LA ORGANIZACIÓN

Por definición, la organización es un sistema conformado por elementos, tales como: Gerencia, Relaciones Industriales, Finanzas, Producción (Servicios), Mercadeo y Ventas; que se unen (se relacionan), para alcanzar los objetivos de la organización (Builes, Carlos, 2003). Gráficamente:

LA ORGANIZACIÓN

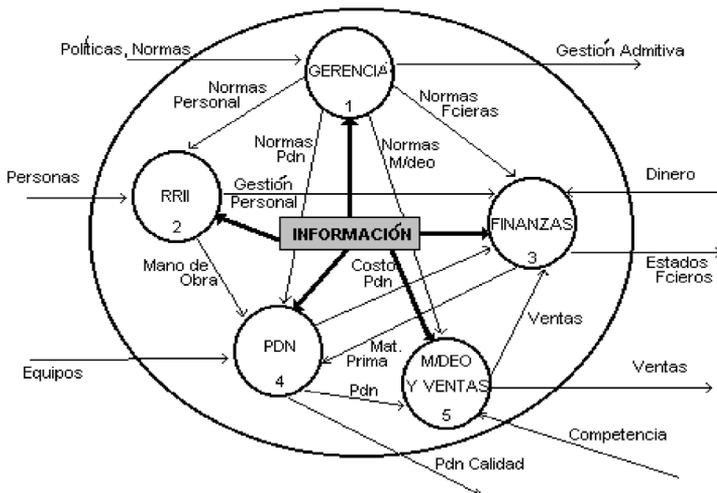


FIGURA 1. LA ORGANIZACIÓN

De esta manera, podemos decir que la organización está influenciada por el medio ambiente y a su vez ésta misma lo modifica.

El medio ambiente en donde se desenvuelve la organización, está conformado básicamente por sistema político, sistema económico, sistema cultural, la competencia, los accionistas, los sindicatos, los proveedores, los clientes y los acreedores.

1.3 ENFOQUE DE SISTEMAS

PARÉNTESIS CONCEPTUAL

Antes se presentarán algunos referentes conceptuales para mayor comprensión del enfoque de sistemas. (Laudon Kenneth, 1994)

- **Sistematización:** Acción y efecto de estructurar, organizar algo con un conjunto ordenado de normas y procedimientos acerca de una determinada materia.
- **Automatización:** Supresión total o parcial de la intervención humana en la ejecución de diversas tareas (sin.: automoción).
- **Computarización:** Someter datos al tratamiento de un computador.
- **Informatización:** Acción y efecto de dotar un servicio, organismo, etcétera, de medios informáticos, asegurar su gestión mediante medios informáticos. Utilizar la informática para tratar con ayuda del computador las necesidades de un sector profesional, para solucionar un problema.
- **Informacionalización:** Representa la importancia creciente de la información, y su explotación, como recurso económico.
- **Cultura de la información:** Una sociedad constituida por ciudadanos y organizaciones informacionalmente cultas (orden espontáneo).
- **Economía de la información:** Economía en la que se ha desarrollado un sector información que contribuya de forma relevante a su crecimiento (planificable).

- Infraestructura: abarca la Economía de la información, es decir, una industria potente en el sector de la información (contenidos, distribución, proceso de información).

Como vemos, en la organización existe un elemento importante que relaciona o vincula a los demás componentes y les proporciona la materia prima para lograr la consecución de sus objetivos particulares, disminuyendo la incertidumbre. Dicho elemento es la información.

Como elemento importante surge, entonces, una nueva manera de mirar la organización, mediante una serie de herramientas que le proporcionan a ésta una metodología de trabajo para resolver problemas, utilizando el concepto de sistema.

Una definición de problema podría ser:

“Un problema existe si hay una diferencia entre lo que está sucediendo actualmente y lo que se desea que suceda. Si no puede decir lo que le gustaría que estuviera sucediendo, todavía no tiene un problema. Simplemente se está quejando.”

Kenneth Blanchard y Spencer Johnson

CARACTERÍSTICAS DEL ENFOQUE

- Es una manera más de mirar los fenómenos del mundo exterior
- Proporciona una visión integral de la realidad (plano macro y micro)
- Hace énfasis en las relaciones entre elementos y de éstos con el medio ambiente
- Cada elemento no importa por SÍ MISMO, sino por el papel que desempeña en el sistema
- La función del sistema NO es igual a la suma de las funciones individuales de los elementos (sinergia)

1.4 SISTEMAS DE INFORMACIÓN

1.4.1 DEFINICIONES

Algunas definiciones acerca de: información, podrían ser:

- La información es la respuesta a una pregunta previamente planteada, la cual disminuye la incertidumbre y proporciona elementos para la toma de decisiones
- La información debe ser vista como un recurso, no como un producto del procesamiento de datos
- La información se ve como un recurso de toda la organización, no sólo del departamento o proyecto que la genera o recibe
- La información proviene de varias fuentes, no sólo de las actividades tradicionales del procesamiento de datos

Algunas definiciones acerca de sistemas de información, podrían ser:

- Un sistema de información es un sistema integrado usuario-máquina/usuario-usuario, que provee información que es de apoyo a las operaciones, la administración y las funciones de toma de decisiones y de control, en una empresa o proyecto determinado, de acuerdo con su planteamiento o estrategia del negocio.
- Es un sistema, cuya función consiste en manejar y administrar el recurso de la información, con el fin de integrar cada uno de los elementos constitutivos de una organización.

Algunos elementos de la organización son:

- Dinero (finanzas)
- Materiales (producción)
- Instalaciones (servicios generales)
- Personal (relaciones industriales)
- Información (sistemas)
- Gerencia (toma de decisiones)

Los sistemas de información son básicos en cualquier actividad que implique la toma de decisiones. Permiten disminuir la incertidumbre, controlando las variables que influyen en el proceso de toma de decisiones.

1.4.2 ELEMENTOS

- Métodos y procedimientos: Define las instrucciones detalladas para delinear las obligaciones, responsabilidades y operación del sistema. Se parte de la Organización para la cual se va a desarrollar el SIBTIC
- Equipo (hardware): Equipos de cómputo
- Personal: De sistemas, usuarios, auditoría y métodos y procedimientos
- Aplicativo (software): Se compone de los diferentes programas de computador que permiten la elaboración de datos e información, de forma más rápida, utilizando el computador
- Bases de datos: Son los almacenamientos de datos de forma estructurada y organizada
- Metodología y documentación: Son las formas estandarizadas para el desarrollo del sistema de información; éstas se componen de herramientas y técnicas. La documentación debe estar relacionada con cada uno de los componentes del sistema de información

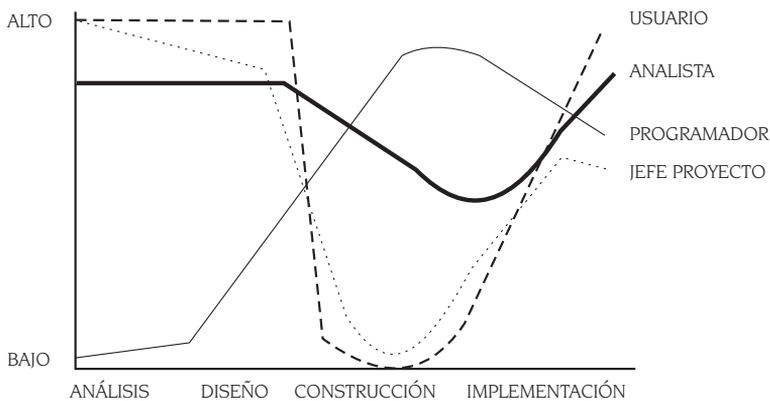


FIGURA 2. GRADO DE PARTICIPACIÓN DE LA PERSONAS EN EL DESARROLLO DE UN PROYECTO*

* Carlos Builes (2003).

Ejemplo: Sistema de Información de Nómina (Builes, 2003).

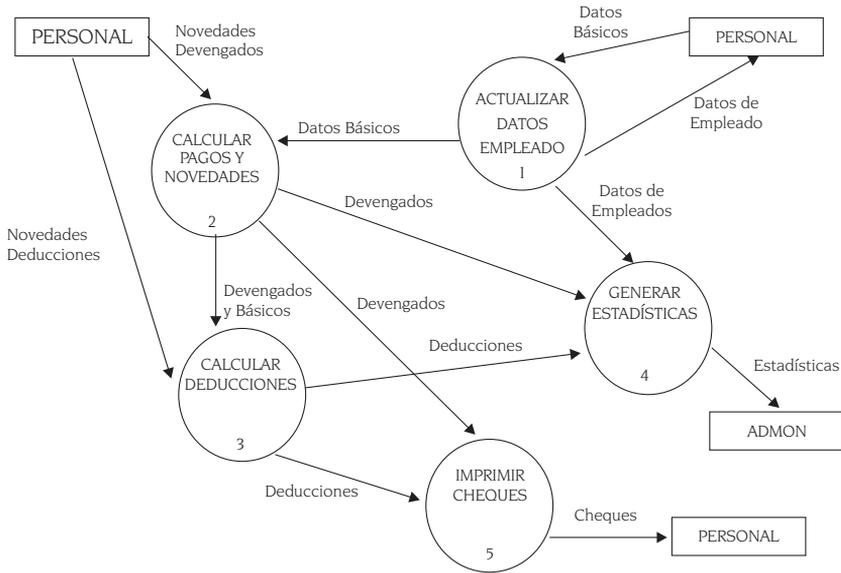


FIGURA 3. SISTEMA DE INFORMACIÓN DE NÓMINA

De la gráfica podemos apreciar que el sistema de información de nómina consta de los siguientes elementos:

Pagos y novedades, manejo de los datos del empleado, manejo de deducciones, generación de estadísticas e impresión de cheques. Todos ellos interactuando (a través de datos e información), para lograr el objetivo básico de generar la nómina para el personal de una organización específica.

1.4.3 PAPEL DE LOS SISTEMAS DE INFORMACIÓN EN LAS ORGANIZACIONES

LEY DE LA ESPECIALIZACIÓN

"Entre más adaptado se encuentre un organismo a un ambiente específico, más difícil le será adaptarse a un ambiente diferente".

La organización conforma un gran sistema en el cual sobresalen dos cualidades:

- Tiene un mecanismo de evaluación o autocontrol (Retroalimentación)
- Es abierto; recibe gran influencia del medio ambiente

El papel de los sistemas de información es soportar un manejo más ágil y eficaz de las variables externas e internas de la organización, para asistir en la toma de decisiones y retroalimentar información para el control de ella misma

Los sistemas de información deben suministrar datos e información suficiente para cada nivel dentro de la organización:



FIGURA 4. NIVELES EN LA ORGANIZACIÓN

Sistemas transaccionales: Se utilizan para automatizar los procesos operativos. Son sistemas intensivos en entrada de datos y salida de información, ya que a través de ellos se cargan las bases de datos de la organización.

Los cálculos y fórmulas que contienen son, en la mayoría de los casos, simples de entender.

Ejemplo: Facturación, Nómina, Cartera, Ventas, etcétera.

Sistemas tácticos: Son sistemas de apoyo a las decisiones. La información que generan le sirve a los mandos medios y a la alta

gerencia. Suelen ser intensivos en cálculos y escasos en entradas de datos. Son dirigidos a un usuario final.

Ejemplo: Simulaciones, Proyecciones Financieras, etcétera.

Sistemas gerenciales (estratégicos): Suelen ser "In House" (hechos en casa). Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos o en servicios.

Apoyan el proceso de innovación de productos y procesos dentro de la empresa.

Ejemplo: MRP (Manufacturing Resource Planning).

A nivel estratégico, los sistemas de información deben apoyar la toma de decisiones mediante la información consolidada y resumida de los sistemas transaccionales, que se encuentran en el nivel operativo. Para el nivel administrativo, los sistemas deben ser una herramienta para ejecutar las decisiones tomadas por el nivel estratégico (sistemas tácticos).

1.4.4 ADMINISTRACIÓN DE SISTEMAS DE INFORMACIÓN

Administrar los sistemas de información, como un recurso más de la organización, implica planeación, organización, ejecución y control.

Dichos elementos administrativos los proporciona la aplicación de la ingeniería de software, un conjunto de herramientas y tareas que tienen como objetivo la planeación, organización, control y ejecución (puesta en marcha de sistemas de información).

A este punto, surgen preguntas importantes acerca de la computarización y uso de los diferentes sistemas de información en una organización:

¿Por qué no debieran computarizarse algunos sistemas de procesamiento de datos?

Por algunas variables como:

- Costo
- Conveniencia
- Seguridad

- Políticas
- Facilidad de mantenimiento
- Naturaleza de los procesos

¿Por qué se han presentado malos resultados aplicando la tecnología informática en las organizaciones?

- El factor humano (humanware) ha sido descuidado, comparándolo con el hardware y el software. No hay formación
- Los sistemas no son integrados. Son islas de computarización.
- Se produce demasiados datos en las organizaciones, lo cual genera embotellamientos y carga
- Se computarizan sistemas ineficientes, mejorando lo que no debería hacerse
- Los usuarios no participan en el diseño de los sistemas que van a utilizar

¿Por qué la tecnología informática no ha mejorado las medidas de éxito en las organizaciones?

- Los beneficios de las tecnologías informáticas no son inmediatamente visibles
- La inversión en tecnología informática no se retorna en forma financiera, sino en forma intangible, en muchos casos
- El impacto de la tecnología informática es muy poco, sino se hace con cambios en la organización
- El uso de herramientas de desarrollo de software es limitado en las organizaciones
- Los sistemas de información son depurados en producción
- Se entregan sistemas con defectos obvios
- Los sistemas son entregados tarde e incompletos, con desfases de más del 200%
- Más del 84% de los sistemas entregados no logran las expectativas originales

1.5 CONSIDERACIONES DE SOFTWARE

1.5.1 DEFINICIÓN

Es la especificación, diseño, programación, prueba, implementación y operación de SIBTIC (Sistemas De Información Basados en Tecnologías de Información Computarizadas).

Es el hábil y disciplinado uso de herramientas de desarrollo de software, con el fin de producir SIBTIC:

Principales características:

Funcionales, económicos, flexibles, confiables, eficientes, fáciles de entender y de modificar, extensibles, seguros, de respuesta rápida, abiertos para comunicarse con otros sistemas.

Se debe usar entonces esta tecnología de software, para aumentar la efectividad (hacer las cosas correctas) más que para aplicar la eficiencia (hacer las cosas correctamente). Dado que computarizar tareas innecesarias aminora la productividad de la organización.

1.5.2 EVOLUCIÓN

La evolución del software se ha motivado por el avance y desarrollo en el hardware. El desarrollo del software se puede agrupar en cinco generaciones:

Primera generación: Sistemas Batch, no portables, monoprogamación. Años: 50 - 60.

Segunda generación: Multiprogramación, sistemas en línea, más portables, bases de datos. Años: 60 - mediados 70.

Tercera generación: Sistemas distribuidos, comunicación entre computadores, mayor complejidad de bases de datos. Años: mediados 70 - principios 80.

Cuarta generación: Micros con mucha velocidad de proceso, gran capacidad de almacenamiento, facilidad de desarrollo. Años: principios 80 - fines 80.

Quinta generación: Inteligencia artificial, objetos, programación parametrizada, Cliente/Servidor, Internet, Intranet. Años: fines 80 - hoy.

Una división en categorías más útil de los sistemas automatizados es la siguiente:

- En línea
- En tiempo real
- Batch
- De apoyo a decisiones
- Basados en el conocimiento

1.5.3 CRISIS DEL SOFTWARE

Se origina aproximadamente entre 1965-1967 con la aparición de los equipos IBM 370.

Debido al desarrollo acelerado del hardware, que permitió realizar múltiples sistemas en un momento, donde no había personal capacitado para hacerlo.

Existían herramientas de hardware potentes para la época, pero el desarrollo de sistemas computarizados era artesanal, ocasionando capacidad ociosa del hardware instalado en las organizaciones. Algunas causas de esta crisis fueron:

- Costos: alto costo de desarrollo, comparado con el costo del hardware
- Escasez del recurso humano capacitado
- Insatisfacción del usuario
- Resultados diferentes a los deseados
- Técnicas de desarrollo muy variadas
- Documentación tediosa
- Mantenimiento de sistemas que consumía mucho tiempo
- Retrasos en los desarrollos

Esta crisis se trata de solucionar mediante el uso de herramientas que ayuden al desarrollo del software.

Surge aquí la INGENIERÍA DE SOFTWARE, como una metodología de trabajo clara para desarrollar SIBTIC (Sistemas de Información Basados en Tecnologías de Información Computarizadas) con una base técnica apropiada.

1.5.4 FACTORES CRÍTICOS DE ÉXITO EN EL DESARROLLO DE SISTEMAS DE INFORMACIÓN

- Identificar el verdadero problema
- Definir claramente los objetivos del proyecto
- Determinar el alcance del proyecto
- Elegir el equipo de trabajo, involucrando tanto a los gerentes como a los usuarios
- Obtener el apoyo de la alta dirección
- Planear las actividades del proyecto
- Obtener los recursos suficientes
- Mantener una cartera balanceada de proyectos de sistemas
- Establecer un sistema de administración de proyectos
- Usar los mecanismos apropiados para ejercer un adecuado control del proyecto
- Tomar en cuenta la evolución de los sistemas
- Garantizar adecuados canales de comunicación
- Asegurar la conformidad con los estándares

1.6 CICLO DE VIDA DE UN SISTEMA DE INFORMACIÓN (SIBTIC)

"El manual del ciclo de vida del proyecto suele ser un libro tan voluminoso como el compendio de normas, que yace (usualmente sin ser leído) sobre el escritorio de todo analista y programador".

Edward Yourdon

Dada la veracidad de la afirmación anterior, ¿de qué sirve, entonces, tener un ciclo de vida de un proyecto?

Existen tres objetivos principales:

1. Definir las actividades a llevarse a cabo en un proyecto de desarrollo de sistemas
2. Lograr congruencia entre la multitud de proyectos de desarrollo de sistemas en una misma organización
3. Proporcionar puntos de control y revisión administrativos de las decisiones sobre continuar o no con un proyecto

Las fases o etapas del ciclo de vida de desarrollo de un SIBTIC son:

- Preanálisis (anteproyecto)
- Análisis
- Diseño
- Construcción
- Pruebas
- Puesta en marcha (entrega al usuario)

1.6.1 PREANÁLISIS O ANTEPROYECTO

Cobija las actividades necesarias para determinar si un sistema particular será desarrollado.

1.6.2 ANÁLISIS

Es la colección, organización y evaluación de HECHOS de un sistema y del medio ambiente en el cual opera, con el objetivo de establecer las BASES de un nuevo sistema SIBTIC, responde a las preguntas:

¿Qué es lo que hace el sistema SIBTIC?

¿Qué es lo que va a hacer el nuevo SIBTIC?

1.6.3 DISEÑO

Es la especificación de la concepción y estructura del nuevo sistema, donde se definen sus módulos, a nivel general y detallado. Responde a la pregunta:

¿Cómo se va a hacer el nuevo sistema?

1.6.4 CONSTRUCCIÓN

Es la programación en un lenguaje específico de sistemas, del diseño antes realizado.

Traduce el diseño a herramientas de sistemas.

1.6.5 PRUEBAS

Son las pruebas funcionales y operacionales que se deben realizar al SIBTIC para verificar la calidad del sistema.

Las pruebas funcionales son las que debe realizar el usuario para verificar que se cumplen los objetivos y los requerimientos planteados al inicio del proyecto y reafirmados en la etapa del análisis.

Las pruebas operacionales son las que deben realizar en el área de sistemas para verificar la calidad del diseño de las bases de datos y la calidad de los programas desarrollados. Éstas verifican la calidad de las técnicas y métodos aplicados.

1.6.6 IMPLEMENTACIÓN (PUESTA EN MARCHA DEL SISTEMA)

Es la validación del nuevo sistema con datos ficticios, para luego colocarlo a funcionar realmente, con datos verídicos.

Los usuarios del sistema deben capacitarse y recibir inducción.

2. EL PROCESO DE DESARROLLO DEL SOFTWARE

“Con el software, al igual que el capital, es el conocimiento incorporado, y puesto que el conocimiento esta inicialmente disperso, el desarrollo del software implícito, latente e incompleto en gran medida, es un proceso social de aprendizaje. El proceso es un dialogo en el que se reúne el conocimiento y se incluye en el software para convertirse en software. El proceso proporciona una interacción entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo (La tecnología). ES un proceso interactivo donde la herramienta de desarrollo se usa como medio de comunicación, con cada iteración del diálogo se obtiene mayor conocimiento de las personas involucradas”.

Howard Batjer, Jr.

El proceso es un conjunto de tareas que se deben llevar a cabo de una forma metódica para desarrollar software de calidad.

Un proceso de software define el enfoque que se toma cuando el software es tratado por la ingeniería. Pero la ingeniería del software también comprende las tecnologías que tiene el proceso –métodos técnicos y herramientas automatizadas.

¿Quién lo hace?

Los ingenieros de software y sus gestores adaptan el proceso a sus necesidades y lo siguen. Las personas que han solicitado el software tienen un papel importante en el proceso del mismo.

¿Por qué es importante?

Proporciona estabilidad, control y organización a una actividad que si no se controla desde su comienzo tiende a volverse caótica.

¿Cómo estar seguro de que lo he hecho correctamente?

La calidad, oportunidad y viabilidad a largo plazo del producto que se esta construyendo son los mejores indicadores de la eficiencia del proceso que estamos utilizando.

Se define un proceso de software como un marco de trabajo de las tareas que se requieren para construir software de alta calidad.

2.1 LA INGENIERÍA DEL SOFTWARE: UNA TECNOLOGÍA ESTRATIFICADA

"La ingeniería del software es la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software".

(Boehm, B.1996)

La ingeniería del software es una tecnología multicapas, donde el fundamento es la capa de PROCESO.

El proceso de la ingeniería del software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la ingeniería del software. El proceso define un marco de trabajo para un conjunto de áreas clave de proceso (ACPs).

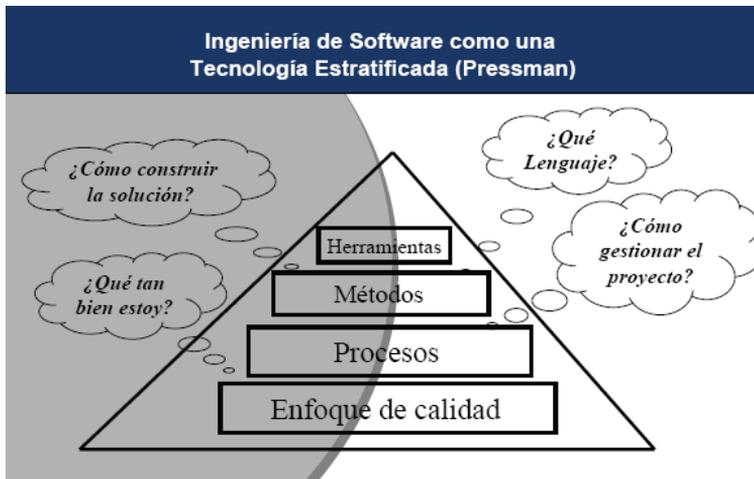


FIGURA 5. INGENIERÍA DEL SOFTWARE COMO UNA TECNOLOGÍA ESTRATIFICADA. (UNIVERSIDAD EAFIT, 2004)

El trabajo que se asocia a la ingeniería del software se puede dividir en tres (3) fases genéricas, con independencia del área de aplicación, tamaño o complejidad del proyecto.

Fase de definición: Intentar identificar qué información ha de ser procesada, qué función y rendimiento se espera, qué compartimien-

to del sistema, interfaces, restricciones de diseño y qué criterios de validación se necesitan para definir un sistema correcto.

Fase de desarrollo: Se define cómo han de diseñarse las estructuras de datos y cómo se traduce este diseño en un lenguaje de programación, se define también cómo hacer pruebas al software.

Fase de mantenimiento: Cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente.

Durante la fase de mantenimiento se encuentran cuatro tipos de cambios (Pressman, 2002):

- Mantenimiento correctivo: Aunque el software se haya desarrollado bien, el cliente puede descubrir algunos defectos. Este cambia los el software para corregir los defectos.
- Mantenimiento de adaptación: Con el paso del tiempo, es probable que cambie el entorno original del software o las reglas de la empresa, es decir, puede cambiar la plataforma tecnológica o políticas de realización de cálculos. Este mantenimiento produce modificación en el software para acomodarlo a los cambios de su entorno externo.
- Mantenimiento de mejora: Conforme se utilice el software, el cliente/usuario puede descubrir funciones adicionales que van a producir beneficios. El mantenimiento de mejora o perfecto, lleva al software más allá de sus requisitos funcionales.
- Mantenimiento preventivo: El software de computadora se deteriora debido al cambio, y por esto el mantenimiento preventivo también llamado reingeniería del software, se debe conducir a permitir que el software sirva para las necesidades de los usuarios finales. En esencia, de computadora, a fin de que se puedan corregir, adaptar y mejorar más fácilmente.

Además de estas actividades de mantenimiento, los usuarios del software desarrollado requieren un mantenimiento continuo. Los asientes técnicos a distancia, teléfonos de ayuda y sitios Web de

aplicaciones específicas, se actualizan frecuentemente como parte de la fase de mantenimiento.

El mantenimiento es mucho más que una simple corrección de errores. Dentro de las actividades se encuentra:

- Seguimiento y control del proyecto de software
- Revisiones técnicas formales
- Garantía de calidad del software
- Gestión de configuración del software
- Preparación y producción de documentos
- Gestión de reutilización
- Gestión de riesgos

2.2 EL PROCESO DEL SOFTWARE

Se establece un marco común del proceso, definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del software, un número de tareas.

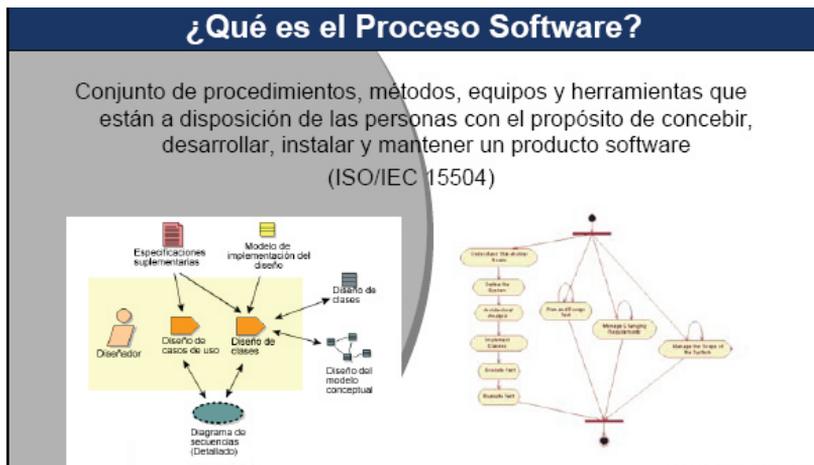


FIGURA 6. PROCESO DE SOFTWARE (UNIVERSIDAD EAFIT, 2004)

2.2.1 CMM UN MODELO PARA LA MEJORA DE PROCESOS DE SOFTWARE

En los últimos años se ha hecho mucho énfasis en la "Madurez del Proceso" (Boehm, B., 1996, pp.73-82). El Software Engineering Institute (SEI) ha desarrollado un modelo completo (CMM - Capability Maturity Model) que se basa en el conjunto de funciones de ingeniería del software que debería estar presente y alcanzar diferentes niveles de madurez del proceso.

CMM describe un camino de mejora para que organizaciones que desarrollan software evolucionen desde un proceso software inmaduro a uno más maduro y disciplinado.

Existen otros modelos de mejora: SPICE, BOOTSTRAP, ISO 9000, PSP.

OBJETIVO DEL CMM

Mejorar la calidad de los procesos de desarrollo, gestión y mantenimiento de software. Para conseguirlo se aplican las bases de la Gestión de la Calidad Total (TQM - Total Quality Management) en la Ingeniería del Software (Según A. Goñi, J.R. Zubizarreta, J. Iturrioz. Dpto. LSI, UPV/EHU; Universidad EAFIT, 2004):

- Mejorar la gestión de la calidad es, en gran medida, responsabilidad de la dirección
- La mejora de la calidad debe basarse en mejorar los procesos y no en las personas
- Hay que medir la mejora de la calidad
- Se necesitan incentivos para mantener un esfuerzo de mejora
- La mejora de la calidad es un proceso continuo

El SEI utiliza un cuestionario de evaluación y un esquema de cinco grados:

NIVEL 1 INICIAL: Se caracteriza según el caso, se definen pocos procesos y el éxito depende del esfuerzo individual.

NIVEL 2 REPETIBLE: Se establecen los procesos de gestión del proyecto para hacer seguimiento del costo.

8. BIBLIOGRAFÍA

- Pressman, Roger S. *Ingeniería del software: un enfoque práctico*. 5ª edición. Madrid: McGraw-Hill, 2002. 601 p
- Senn, James A. *Análisis y diseño de sistemas de información*. 2ª edición. México: McGraw-Hill, 1992. 942 p
- Larman, Craig. *UML y patrones introducción al análisis y diseño orientado a objetos*. 2ª edición México: Prentice-Hall, 1999. 507 p
- McConnell, Steve. *Desarrollo y gestión de proyectos informáticos cómo dominar planificaciones ajustadas de software*. 2ª edición. Madrid: McGraw-Hill, 1997. 691 p
- Kendall, Kenneth E.; KENDALL, Julie E. *Análisis y diseño de sistemas*. 3ª edición. México: Prentice-Hall, 1997. 913 p
- Builes, Carlos "Notas de Ingeniería". (Sin publicar 2003).
- Laudon Kenneth, *Sistemas de información gerencial*. 6º Edición México: McGraw-Hill, 1994. Capítulo 18
- Documentos de EAFIT. *Tertulias sobre Ingeniería del software*. Universidad EAFIT, 2004.
- Boehm, B., *A Spiral Model for Software Development and Enhancement*, Computer, vol. 21, n.Q 5, mayo 1998, pp 61-72.
- Boehm, B., *Anchoring the Software Process*, IEEE Software, vol. 13, n.Q 4, julio 1996, pp.73-82.
- Boehm, B., *Using the WINWIN Spiral Model: A Case Study*, Computer, vol. 31, n. Q 7, julio 1998, pp. 33-44.
- Brooks, F., *The Mythical Man-Month*, Addison-Wesley, 1975.
- McDermid, J. y P. Rook, *Software Development Process Modelw, en Softilwe EngineerS Reference Book*, CRC Press, 1993, pp. 15/26-15/28.
- Nierstrasz, *Komponent-Oriented Software Development*, CACM, vol. 35, n.Q 9, septiembre 1992, pp.160-165.
- Mills, H.D., M. Dyer y R. Linger, *Clearroom Software Engineerinng*, IEEE Software, septiembre 1987, pp.
- Guerrero, Diego. SIARI "Sistematización y computarización de la Información de Administrativa de Recursos Humanos y Físicos de la Fundación Fe y

- Alegría (Regional Antioquia)". Proyecto de investigación ITM 2004.
- Cause, D.C., y G. M. Weinberg, *Exploring Requirements: Quality Before Design*, Dorset House, 1989.
- Akao, Y. (de.), *QualiQ Function Deployment: Integrating Customer Requirements in Product Design* (traducido por G. Mazur), Productivity Press, Cambridge MA, 1990.
- Zultner, R., *Quality Function Deployment for Software: Satisfying Customers*, *American Programmer*, febrero 1992, pp. 28-41.
- Jeff Johnson. "Gui Bloopers: Don'ts and Do's for Software Developers and Web Designers". Morgan Kaufman. 2000. <http://web-bloopers.com/gui-bloopers/figures.php>

REFERENCIAS ADICIONALES

Sitio con temas varios acerca del diseño de objetos. Referencia a diversas herramientas CASE con UML: http://www.objectsbydesign.com/Formalizando_UML <http://www.iist.unu.edu/home/Unuiist/newrh/III/1/docs/techreports/report230.pdf>

Integrating User Interface Design And Object-Oriented Development Through Task Analysis And Use Cases. John M. Artin. Disponible en:

- <http://www.cutsys.com/CHI97/Artim.html>
- <http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/default.asp>
- <http://www.cic.ipn.mx/ute/ogonzalez/Cursos/tp01/sld001.htm>
- http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public_html/
- www.baddesign.com