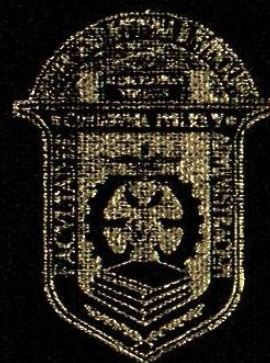
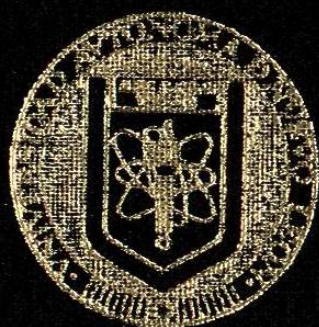


UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CONTADURIA PUBLICA
Y ADMINISTRACION

DIVISION DE POST-GRADO



ESTRATEGIAS PARA LOGRAR EL DESARROLLO E
IMPLANTACION DE UN SISTEMA DE INFORMACION
CON CALIDAD EN LOS NEGOCIOS

T E S I S

PRESENTADA COMO REQUISITO
PARA OBTENER EL GRADO ACADEMICO DE
MAESTRO EN INFORMATICA ADMINISTRATIVA

QUE PRESENTA:

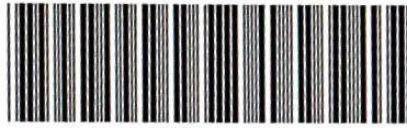
MARIA DEL CONSUELO JIMENEZ FERNANDEZ

MONTERREY, N. L.

AGOSTO DE 1994

ESTRATEGIAS PARA LOGRAR EL DESARROLLO E
IMPLANTACION DE UN SISTEMA DE INFORMACION
CON CALIDAD EN LOS NEGOCIOS

TM
Z7164
.C8
FCPYA
1994
J5



1020073607

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE CONTADURIA PUBLICA Y ADMINISTRACION

DIVISION DE POSTGRADO

**Estrategias para lograr el desarrollo e implantación
de un sistema de información con calidad en los
negocios**

TESIS

**PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO
ACADEMICO DE:
MAESTRO EN INFORMATICA ADMINISTRATIVA**

QUE PRESENTA

MARIA DEL CONSUELO JIMENEZ FERNANDEZ

MONTERREY, N.L.

AGOSTO DE 1994

TM
Z71 4
.C8
FCP4A
1994
J5



FONDO TESIS

62787

DEDICATORIAS

A Dios a quien le debo todo lo que soy, gracias por haberme dado la fuerza y salud para seguir adelante.

A mi esposo Rafael por la paciencia, apoyo, comprensión y amor que me ha brindado, ya que gracias a él he podido lograr una meta más de mi vida.

A mi madre le doy las gracias por la educación que me brindó y porque ha sido y continuará siendo la base de mi vida, por sus consejos y su apoyo incondicional.

Y a todas aquellas personas que con sus palabras de aliento me han motivado a salir adelante.

AGRADECIMIENTOS

Al Lic. Enrique Hernández Hernández

Al Ing. José Humberto Martínez Jiménez

y a la Ing. Leticia Neira Tovar

Por su apoyo, tiempo y conocimientos brindados para el desarrollo de esta tesis.

A todos mis maestros por haber compartido sus conocimientos.

A la Universidad de Monterrey por haber hecho posible la realización de mis estudios de Maestría.

A mi ex-jefe Ing. Luis Bernardo Gordillo por toda la ayuda proporcionada durante mis estudios de Maestría.

A la Lic. Ana. Ma. Piña, Lic. Laura Tovar, Ing. Liliana González y Francisca Bustos gracias por su disposición y ayuda, ya que gracias a ustedes pude tener acceso a mucha de la información que fué fundamental para esta tesis.

A la Lic. Dolores Garza por su valiosa ayuda y tiempo dedicado a la revisión de este documento.

A todos mis alumnos en especial a

Nelda, Alejandra, Juan Ramón, Luis Manuel, Irma Nelly, Claudia, Paula, Rosa Nelly, Oscar, Ricardo, Adrián, Alejandro y muchos más que en estos últimos años han colaborado conmigo en muchos proyectos sin esperar nada a cambio, y que gracias a ellos tuve el deseo de ser cada vez mejor y de seguirse superando por ellos y para ellos.

INDICE GENERAL

Resumen.....	XIII
Introducción	
Antecedentes.....	XIV
Objetivos.....	XIV
Alcances.....	XIV
Beneficios.....	XV
1.El desarrollo de los sistemas de información	
1.1 Definición.....	1
1.2 Componentes básicos.....	1
1.3 Ciclo de vida de los sistemas de información.....	2
1.4 El ciclo de vida del desarrollo de software en la actualidad.....	2
1.4.1 Ciclo de vida clásico.....	3
1.4.2 Ciclo de vida del desarrollo con prototipos.....	5
1.4.3 Ciclo del desarrollo con herramientas de cuarta generación.....	6
1.5 La evolución en el desarrollo del software.....	10
1.6 Medio ambiente del desarrollo de software.....	12
2.Calidad.	
2.1 Definición.....	15

2.2 Desarrollo histórico hacia la calidad.....	15
2.3 Filosofías y técnicas de calidad	
2.3.1 Filosofías.....	17
2.3.2 Los expertos y su calidad total.....	22
2.3.3 Técnicas.....	26
2.4 Calidad y productividad.....	29
2.5 En busca de la calidad:capacitación.....	29

3. Metodologías, técnicas y herramientas que mejor soportan el desarrollo de un sistema.

3.1 Análisis y diseño estructurado.....	31
3.2 Ingeniería de software.....	32
3.3 Ingeniería de información.....	34
3.4 RAD.....	38
3.5 Prototipo.	
3.5.1 Definición.....	41
3.5.2 Tipos.....	42
3.5.3 Usos.....	42
3.5.4 Las herramientas para prototipos.....	43
3.5.5 Beneficios y peligros.....	44
3.6 JAD.....	44

4. CASE.

4.1 Componentes.....	47
4.2 Clasificación.....	49

4.3 Herramientas actuales en el mercado internacional.....	53
4.4 Ventajas y desventajas.....	54
4.5 La implementación del CASE.....	55
4.6 Características deseables para una metodología de desarrollo de sistemas.....	65
4.7 Elementos indispensables para la selección de CASE.....	66
4.8 CASE y la quinta generación.....	71
4.9 Ciclo de vida tradicional contra el ciclo de vida con CASE.....	74
4.10 CASE en algunas empresas de Monterrey.....	77
4.11 Perfil de las empresas de Monterrey y su área metropolitana.....	83

5. Calidad en el desarrollo de sistemas.

5.1 Sistemas: un departamento de producción.....	97
5.2 Calidad en el software.....	99
5.3 Hacia la calidad sus determinantes y factores.....	102
5.4 Los factores de la calidad	103
5.5 Métricas.....	106
5.6 Estándares.....	109
5.7 Inspección o prevención.....	113
5.8 El factor humano.....	115
5.9 Modelo propuesto.....	120

6. Guía y estrategias para desarrollar e implantar un sistema con calidad.....	124
---	------------

Conclusiones.....174

Bibliografía.

Bibliografía de referencias.....176

Bibliografía general.....179

Anexos.

Anexo A. Herramientas CASE.....185

Anexo B. Formato de cuestionarios aplicados a empresas para
conocer tanto el perfil en lo que respecta al área de desarrollo
sistemas así como su conocimiento y uso de herramientas
CASE.....193

Anexo C. Formatos de solicitud de servicio y mantenimiento a
sistemas.....209

Anexo D. Distribución del tiempo dedicado al desarrollo de sistemas
con herramientas CASE y sin ellas.....211

Anexo E. Compañías en Estados Unidos que actualmente ofrecen
excelentes servicios en cuanto a tecnología CASE se refiere.213

Anexo F. Herramientas para lograr la calidad.....215

INDICE DE FIGURAS

Fig.1.1 Diagrama de proceso que muestra la interacción entre los componentes del sistema.....	1
Fig.1.2 Ciclo de vida de un sistema de información.....	2
Fig.1.3 Ciclo de vida clásico.....	3
Fig.1.4 Modelo del ciclo de vida tradicional.....	4
Fig.1.5 Ciclo de vida de desarrollo con prototipos.....	5
Fig.1.6 Ciclo de desarrollo con 4GL.....	6
Fig.1.7 Combinación de enfoques.....	7
Fig.1.8 La re-ingeniería en el software.....	8
Fig.1.9 La re-ingeniería y el CASE.....	9
Fig.1.10 CASE y su historia.....	13
Fig.2.1 El desarrollo del control de calidad.....	16
Fig.3.1 Representación gráfica del modelo de contexto.....	31
Fig.3.2 Beneficios con las herramientas que soportan el análisis y diseño estructurado.....	32
Fig.3.3 La ingeniería de software en el desarrollo de sistemas.....	33
Fig.3.4 Actividades a seguir para la implementación de la ingeniería de software.....	34
Fig.3.5 Lo que representa la ingeniería de información.....	37
Fig.3.6 Etapas en el desarrollo rápido de aplicaciones.....	38
Fig.3.7 Ciclo de vida clásico vs. RAD.....	39
Fig.3.8 Elementos indispensables para aplicar RAD.....	40

Fig.3.9 Fases del JAD y sus beneficios.....	46
Fig.4.1 Componentes de las herramientas CASE.....	49
Fig.4.2 El modelo de etapas para implementar CASE según Hugues y Clark.....	58
Fig.4.3 El modelo de etapas y su relación con las funciones de una metodología.....	59
Fig.4.4 Involucramiento del personal de sistemas en las etapas de implementación de CASE.....	60
Fig.4.5 La organización en las etapas de implementación de CASE.....	61
Fig.4.6 Proceso para asimilar CASE.....	62
Fig.4.7 Elementos indispensables para la selección de CASE.....	66
Fig.4.8 Componentes del conductor de la metodología en una herramienta CASE.....	73
Fig.4.9 Porcentaje de recursos utilizados en el ciclo de vida tradicional de un sistema.....	74
Fig.4.10 Porcentaje de recursos utilizados en el ciclo de vida del sistema con técnicas estructuradas.....	74
Fig.4.11 Porcentaje de recursos utilizados en el ciclo de vida con CASE.....	75
Fig.4.12 Elementos a considerar para lograr la calidad en un sistema.....	76
Fig.4.13 Tipos de industria.....	84
Fig.4.14 Ingresos brutos por año.....	84
Fig.4.15 Cantidad de empleados en el empresa.....	85
Fig.4.16 Cantidad de empleados en el departamento de informática.....	85
Fig.4.17 Cantidad de empleados destinados al desarrollo de sistemas.....	86
Fig.4.18 El sistema operativo para el medio ambiente de producción.....	86
Fig.4.19 El sistema operativo para el medio ambiente de desarrollo.....	87

Fig.4.20 Principales sistemas administradores de bases de datos.....	87
Fig.4.21 Principales lenguajes de desarrollo.....	88
Fig.4.22 Proporción de PC's, estaciones de trabajo o terminales para los desarrolladores	88
Fig.4.23 Lo más importante para la organización.....	89
Fig.4.24 Características de un programa de calidad utilizado en las empresas.....	89
Fig.4.25 Filosofía de calidad que las empresas utilizan en el desarrollo de sistemas.....	90
Fig.4.26 Uso de metodologías para el desarrollo de sistemas-métodos estructurados.....	90
Fig.4.27 Uso de metodologías para el desarrollo de sistemas-metodologías del ciclo de vida.....	91
Fig.4.28 Experiencias en relación a las herramientas CASE.....	91
Fig.4.29 Enfoques para mejorar el proceso de desarrollo en la organización....	92
Fig.4.30 Principales obstáculos para adoptar CASE en la organización.....	93
Fig.4.31 Recomendación de las empresas para implantar CASE.....	94
Fig.4.32 Empresas cuyas aplicaciones han sido desarrolladas por herramientas CASE.....	94
Fig.4.33 Tipo de capacitación que la empresa recibe.....	95
Fig.5.1 Relación entre el departamento de sistemas y el de producción.....	97
Fig 5.2 El control de calidad como elemento indispensable en el desarrollo de sistemas.....	98
Fig.5.3 Los objetivos de calidad en el área de desarrollo de sistemas.....	101
Fig.5.4 Elementos indispensables para dar valor a los procesos del cliente....	101
Fig.5.5 Determinantes de la calidad.....	102

Fig.5.6. Las métricas como medidores de la presencia de los factores de
calidad..... 106

Fig.5.7 Diagrama necesario para medir la complejidad en un programa..... 107

Fig.5.8 La calidad por inspección..... 114

Fig.5.9 La calidad por prevención básica en el desarrollo e implementación
de un sistema..... 115

Fig.5.10 Determinantes y sus componentes para el logro de un sistema con
calidad..... 120

Fig.6.1 Estrategias para el logro de la calidad..... 173

INDICE DE TABLAS

Tabla 4.1 Herramientas que soportan todo el ciclo de vida del desarrollo de un sistema.....	53
Tabla 4.2 Desarrollo tradicional vs. desarrollo con CASE.....	75
Tabla 4.3 CASE en empresas que pertenecen a la industria del vidrio.....	77
Tabla 4.4 CASE en empresas que pertenecen a la industria de celulosa y derivados.....	78
Tabla 4.5 CASE en empresas que pertenecen a la industria de productos alimenticios.....	78
Tabla 4.6 CASE en instituciones financieras.....	80
Tabla 4.7 CASE en empresas que pertenecen a la industria de maquinaria e implementos agrícolas.....	80
Tabla 5.1. Los factores de calidad.....	104
Tabla 5.2 Relación entre los Factores de McCall y Freeman.....	105
Tabla 5.3 Relación entre las métricas y los factores de calidad.....	109
Tabla 5.4 Estándar DOD con enfoque al factor funcionalidad.....	110
Tabla 5.5 Estándar FAA con un enfoque al aseguramiento de calidad.....	110
Tabla 5.6 Estándar IEEE con un enfoque a diversos factores de calidad.....	111
Tabla 5.7 Estándar ISO-9004 con un enfoque a la calidad en el proceso y el producto.....	113
Tabla 5.8 Personal necesario en el desarrollo de un sistema utilizando herramientas CASE u otro tipo de herramientas.....	119

RESUMEN

La problemática que en estos últimos años ha existido en el área de desarrollo de sistemas ha ocasionado que los sistemas de información no sólo no funcionen adecuadamente sino que no se entreguen a tiempo y no cumplan con los requerimientos y peticiones de los usuarios, ocasionando después de su instalación excesivos mantenimientos, logrando con esto invertir más recursos en la modificación de los sistemas que en nuevos desarrollos.

Es por eso que a través de esta tesis se pretende dar a conocer todos aquellos factores indispensables a considerar en el proceso de desarrollo de un sistema, proceso que debe estar firmemente apoyado por toda una filosofía no sólo de calidad, sino además por una que encierre a las metodologías, técnicas y herramientas como lo es la filosofía CASE. Lo anterior, es siendo todo esto necesario para desarrollar e implantar un sistema con calidad

Si conocemos todos estos factores y tomamos en cuenta la guía así como las estrategias propuestas, se podrán prevenir aquellos problemas que hasta la fecha se han presentado en ese proceso de desarrollo tal que el sistema a obtenerse sea un producto adecuado a las necesidades de su cliente.

INTRODUCCION

Antecedentes.

Constantemente el departamento de sistemas es bombardeado con solicitudes de nuevos desarrollos, y de cambios a los sistemas existentes, que requieren ser realizados con presteza.

Pero como por lo general dicho departamento está acostumbrado a no utilizar metodologías o a utilizarlas inadecuadamente ha contribuido a que el proceso de desarrollo de sistemas sea costoso, consume más tiempo que lo esperado e increíblemente retrase el trabajo.

Además el nuevo sistema como sigue teniendo problemas en su diseño, construcción e implantación por ese desarrollo al vapor es un sistema pobre, sin calidad, ya que no cumple con los requerimientos, siendo los errores de éste encontrados por el propio usuario durante su operación.

Esta calidad de la que hablamos no ha sido considerada por la mayoría de las empresas, sin embargo hay que pensar que la calidad ya no es exclusiva de un departamento de producción, hoy y siempre la calidad debe ser parte de toda la empresa. más aún si ésta quiere ser de clase mundial, nuestro reto por lo tanto es ser mejores, hacer las cosas bien, a la primera y hacer mas con menos.

El departamento de sistemas deberá construir un sistema que proporcione la información que el usuario necesite debiendo ser esta considerada por la dirección de las empresas como un factor estratégico para el desarrollo y competitividad dándole así el valor que merece tanto a la información como al departamento.

Por lo tanto si tomamos en consideración la situación actual en que vivimos el departamento de sistemas estará bajo una presión cada vez mayor ya sea por el nivel competitivo que desea alcanzar la empresa, por el recorte de presupuesto, el poco personal, el cambio constante en las estrategias del negocio esto sólo ocasionaría necesidades de nuevos sistemas desarrollados cada vez en menos tiempo. si queremos hacer frente a todos estos problemas la respuesta a esto sera simplemente la calidad

Objetivo.

Dar a conocer la filosofía, estrategias, metodologías, técnicas y herramientas que permitirán desarrollar e implementar un sistema de información con calidad con énfasis en la tecnología CASE. Proponer además la guía que dirija al equipo de desarrollo de sistemas, estableciendo las actividades a desempeñar, las mejores técnicas y herramientas a utilizar, previniendo los problemas a través del conocimiento de estos soportados por una serie de acciones para eliminarlos

Alcances.

- Dar a conocer la historia en cuanto al desarrollo de sistemas y a la calidad se refiere.

- *Presentar aquellas filosofías técnicas y herramientas que se enfocan a la calidad*

- *Indicar las metodologías, técnicas y herramientas específicamente.*

- + Estructurada.
- + Ingeniería de información
- + Prototipo
- + RAD
- + Ingeniería de software
- + JAD
- + CASE.

que pueden soportar el ciclo de vida de desarrollo de un sistema de información

- *Mostrar las herramientas CASE, sus componentes su clasificación, las compañías que las distribuyen, sus ventajas y desventajas, las herramientas CASE más importantes, los diferentes enfoques que se deben tomar en cuenta para su implementación, los factores a considerar para su evaluación, la realidad con respecto a éstas en algunas empresas de nuestra localidad así como lo que se espera de CASE en un futuro.*

- *Dar a conocer la problemática que presenta el área de desarrollo de sistemas en algunas de las empresas más importantes de nuestra localidad.*

- *Definir todos aquellos elementos importantes a considerar para lograr un sistema de información con calidad.*

- *Explicar aquellos factores de calidad que deben tomarse en cuenta para determinar si un sistema tiene calidad así como las métricas relacionadas con éstos.*

- *Dar a conocer el perfil que los integrantes del equipo de desarrollo deben tener para lograr un trabajo de calidad.*

- *Señalar aquellos estándares generados por organizaciones que aseguran la calidad en los sistemas.*

- *Describir la filosofía de calidad propuesta.*

- *Mostrar la guía que permita producir un sistema con calidad involucrando todos los elementos antes mencionados así como técnicas y herramientas que mejor soporten esta guía; prevenir los problemas a través del conocimiento de las actividades a realizar, como de las posibles causas que ocasionan los problemas y dar una solución, para lograr así mejorar el proceso de desarrollo apoyado, además por estrategias de calidad.*

Beneficios.

- *Guiar al desarrollador de sistemas a través de todo el proceso del ciclo de vida del desarrollo de sistemas (CVDS) con el fin de lograr sistemas que satisfagan los requerimientos del usuario y del negocio.*

- *Permitir que el departamento de sistemas conozca cuáles son los requisitos que debe cumplir una herramienta CASE y si su empresa actualmente está preparada para su utilización.*

- Mejorar la imagen del area de desarrollo de sistemas a través del mejoramiento de la calidad en sus productos
- Involucrar al usuario en todo el proceso del CVDS mejorando así la comunicacion
- Lograr no sólo la calidad en los sistemas sino la productividad en el equipo de desarrollo.
- Permitir que las mejoras a los sistemas sean realizadas en el menor tiempo posible y al menor costo
- Y empezar a crear una cultura de calidad y una nueva cultura computacional en el departamento de sistemas

1. EL DESARROLLO DE LOS SISTEMAS DE INFORMACION.

1.1. DEFINICION.

Los sistemas de información son un conjunto ordenado y formal de componentes que realizan operaciones de procesamiento de datos con el fin de

- Satisfacer las necesidades de procesamiento de datos en relación a las transacciones que se llevan a cabo en las organizaciones
- Proporcionar información a los administradores para soportar el proceso de la toma de decisiones

Por el procesamiento de datos, la actualización a los archivos de la empresa y la generación de información, los sistemas de información hacen posible el logro de las metas organizacionales, apoyando a las diferentes actividades y logrando el ahorro de recursos, así como la agilización de procesos.

1.2. COMPONENTES BASICOS.

Los componentes de cualquier sistema de información son

Entradas: Datos que sirven como materia prima para procesamiento

Procesamiento: Incluye las actividades que transforman datos de entrada en información útil.

Salidas: Es el producto o resultado del procesamiento.

Retroalimentación: Se considera como una salida especialmente diseñada y utilizada para verificación, control de calidad y evaluación de resultados.

Control: Existe dentro de cualquier función que prueba la retroalimentación del sistema para determinar si el desempeño es de acuerdo a lo esperado

Ajustes: Son los productos del proceso de control que llevan entradas al sistema o retroceden el procesamiento de acuerdo a lo esperado.

Todos estos componentes trabajan juntos transformando datos de entrada en información de salida para mantener el sistema de procesamiento en niveles aceptables de calidad.

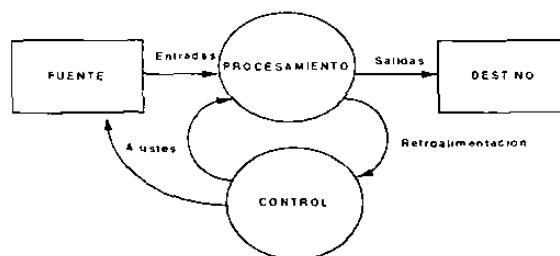


Fig 1.1 Diagrama de proceso que muestra la interacción entre los componentes del sistema

1.3. CICLO DE VIDA DE LOS SISTEMAS DE INFORMACION.

El ciclo de vida de los sistemas de información puede representarse por la siguiente figura

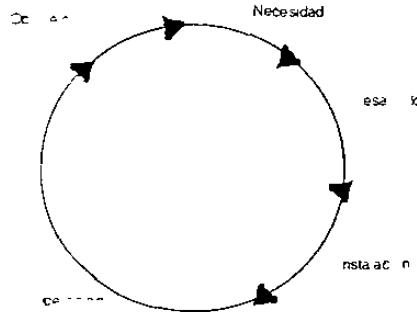


Fig 1.2 Ciclo de vida de un sistema de información

Los constantes cambios que existen en el medio ambiente de negocios motivan a solicitar nuevos sistemas de información o a mejorar los ya existentes.

Esto es tal que si se requiere llevar a cabo el desarrollo de un sistema implicará analizar las necesidades del cliente para diseñarlo e implementarlo, logrando después de ser desarrollado su aceptación para ser instalado.

Luego de su instalación existe un período de uso activo siendo esta la parte más larga del ciclo de vida del sistema, ya que debido a los cambios que puedan surgir en la empresa o en el medio ambiente que la rodea, puede llegar a ser obsoleto reemplazándolo y terminando así su ciclo, o mejorarlo y adaptarlo en base a los nuevos requerimientos.

1.4. EL CICLO DE VIDA DEL DESARROLLO DE SOFTWARE EN LA ACTUALIDAD.

El CVDS se puede definir como un conjunto de actividades requeridas para definir, desarrollar, probar, entregar, operar y mantener un sistema.

Cada vez que se requiere desarrollar un nuevo sistema es necesario definir su CVDS para:

- Definir las actividades que se deben ejecutar para desarrollar el software
- Lograr que el proyecto sea consistente con los demás sistemas computacionales desarrollados en la empresa

- Definir puntos de control donde se puedan tomar decisiones de continuar o suspender el proyecto

En cuanto al ciclo de vida de desarrollo de software podemos citar diferentes enfoques

1.4.1. CICLO DE VIDA CLASICO.

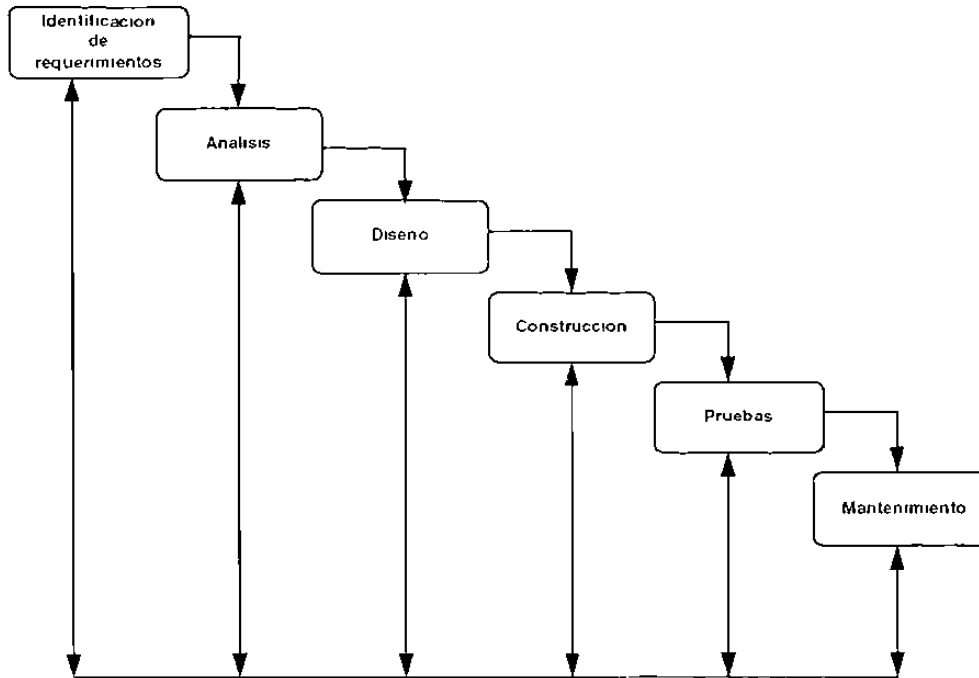


Fig. 1.3 Ciclo de vida clásico

Donde cada fase involucra.

- Identificación de requerimientos.
Consiste en establecer los requisitos de todos los elementos del sistema
- Análisis de los requerimientos de software
Estudio de los requerimientos de datos para poder identificar las características del nuevo sistema
- Diseño
A través del diseño hacemos énfasis a la estructura de los datos, arquitectura del software, al detalle procedimental y a la interfaz, permitiendo traducir los requerimientos en una forma que pueda ser establecida, obteniendo así calidad antes de la codificación

- d) Codificación
Traducir el diseño a una forma legible por a computadora entre mas detallado sea el diseño la codificación sera mecanicamente
- e) Pruebas
Cada uno de os programas que conforman al sistema deberan probarse asegurando que cumplan los requerimientos de los usuarios
- f) Mantenimiento
El software sufrira cambios una vez entregado el sistema ya sea por
 - Errores
 - Por cambios en el entorno externo
 - Necesidades por parte del cliente en relacion a aumentos funcionales o de rendimiento

Se puede observar que las diferentes fases deberan ser realizadas en forma secuencial si se desea producir un sistema Involucrando una mayor cantidad de recursos en la programacion y pruebas, por el tipo de herramientas utilizadas (lenguajes de programacion tradicionales)

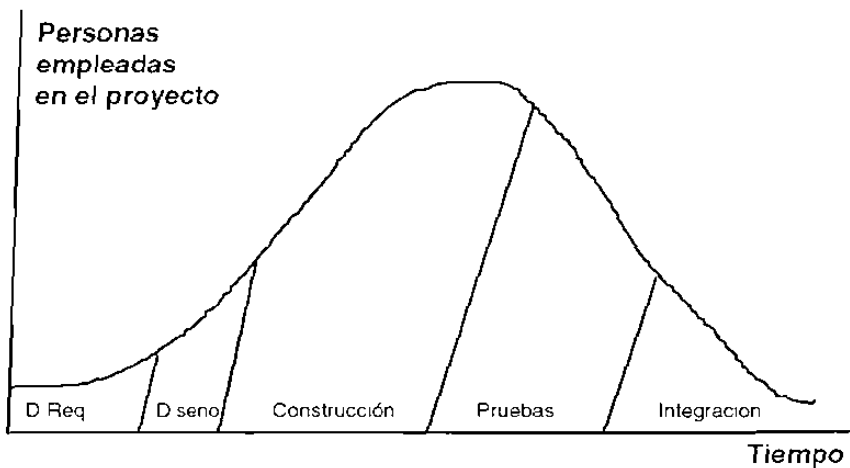


Fig. 1.4 Modelo del ciclo de vida tradicional

1.4.2. CICLO DE VIDA DEL DESARROLLO CON PROTOTIPOS.

Hoy en día el análisis de requerimientos se ve beneficiado por la técnica por muchos de nosotros conocida como prototipo, ya que a través de un proceso iterativo podemos identificar los requerimientos que debe incluir el sistema para satisfacer así las necesidades del usuario. Esto se logra identificando los requerimientos con ayuda del usuario construyendo previamente un modelo con los requerimientos determinados, logrando posteriormente que el usuario trabaje con el prototipo revisando y refinando los requerimientos para posteriormente hacer cambios pertinentes y así hasta que el usuario quede realmente satisfecho.

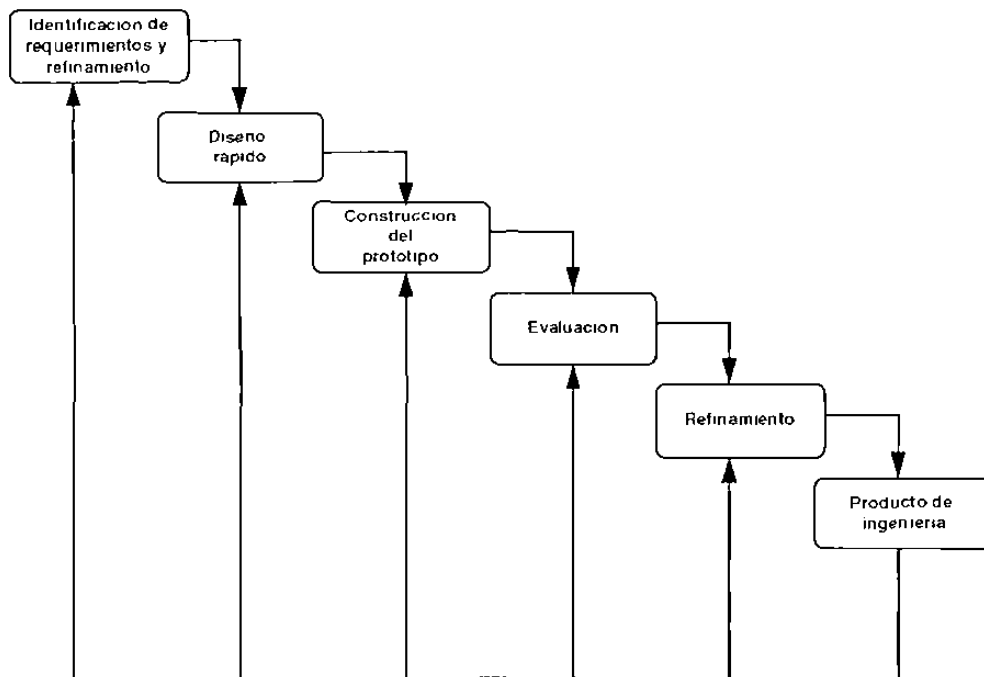


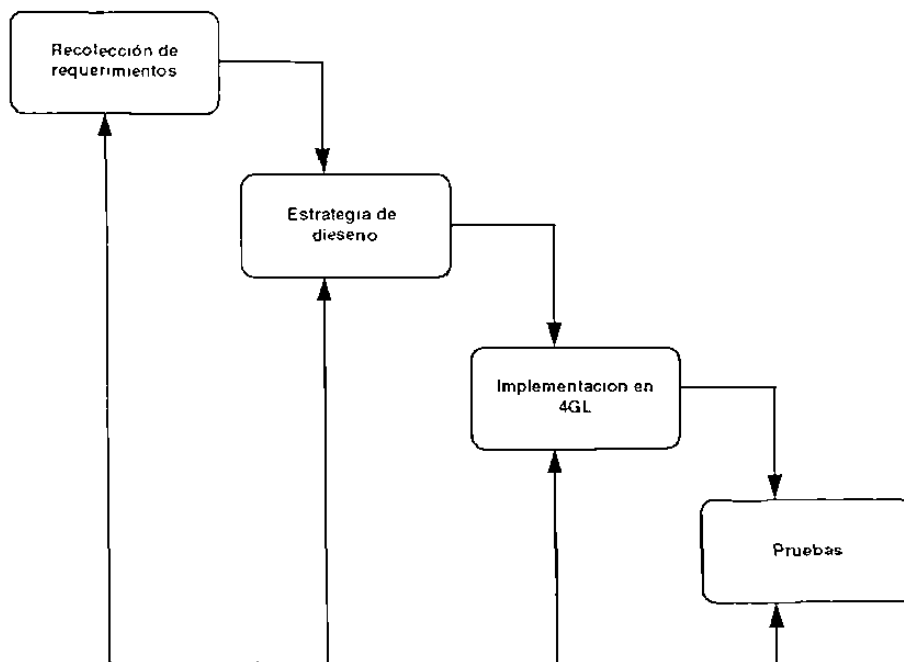
Fig. 1.5 Ciclo de vida de desarrollo con prototipos

- a) Determinar con el usuario los requerimientos para el modelo de aplicación.
- b) Diseño rápido
Con enfoque a entradas y formatos de salidas
- c) Construcción del prototipo
Desarrollo utilizando un lenguaje de cuarta generación.
- d) Evaluar los requerimientos.
Procesar el modelo e identificar iterativamente sus adaptaciones a la situación deseada

- e) Refinar e prototipo
Para que satisfaga las necesidades del usuario logrando que la persona que lo desarrolla comprenda mejor o que ha y que hacer
- f) Producto de ingeniería
Procesar los cambios reacionados con la etapa anterior hasta que la aplicacion sea la deseada

1.4.3. CICLO DEL DESARROLLO CON HERRAMIENTAS DE CUARTA GENERACION.

El desarrollo con herramientas como Focus Oracle Natural, Nomad Powerhouse Ramis entre otras integran un diccionario de datos y dinamicas bases de datos relacionales o de otro tipo, permitiendo al programador especificar sólo el que hacer logrando así reducir los esfuerzos de la programacion Tomando como base lo anterior este desarrollo soportado por 4GL estaria representado como



F g. 1.6 Ciclo de desarrollo con un 4GL.

- a) Recoleccion de requer m entos
El analista de sistemas obtendrá los requerimientos del usuario
- b) Estrategia de diseño
Realizar un diseño detallado para asegurar la calidad de la aplicación así como su estabilidad a través del tiempo

- c) Implementacion en un lenguaje de cuarta generacion
 Utilizar un lenguaje de cuarta generacion el cual debera de acceder una estructura de datos con informacion relevante generando asi programas ejecutables
- d) Pruebas
 Realizar una prueba completa y generar la documentación

Analizando todos estos enfoques podemos concluir que para todo proceso de desarrollo de sistemas independientemente del area de aplicación, tamaño del proyecto o complejidad, será necesario realizar un analisis de requerimientos, desarrollo y mantenimiento. Pero debemos pensar que existen ya muchas tecnicas y herramientas que pueden soportar este proceso de desarrollo y son los desarrolladores quienes deberan seleccionar tanto el enfoque que más les convenga, como las herramientas y tecnicas que mejor lo soporten de acuerdo al tipo de proyecto al que se enfrenten, lo mismo sucederia si se tratara de mejorar un sistema

A continuacion se muestra una combinacion de todos los enfoques anteriormente explicados

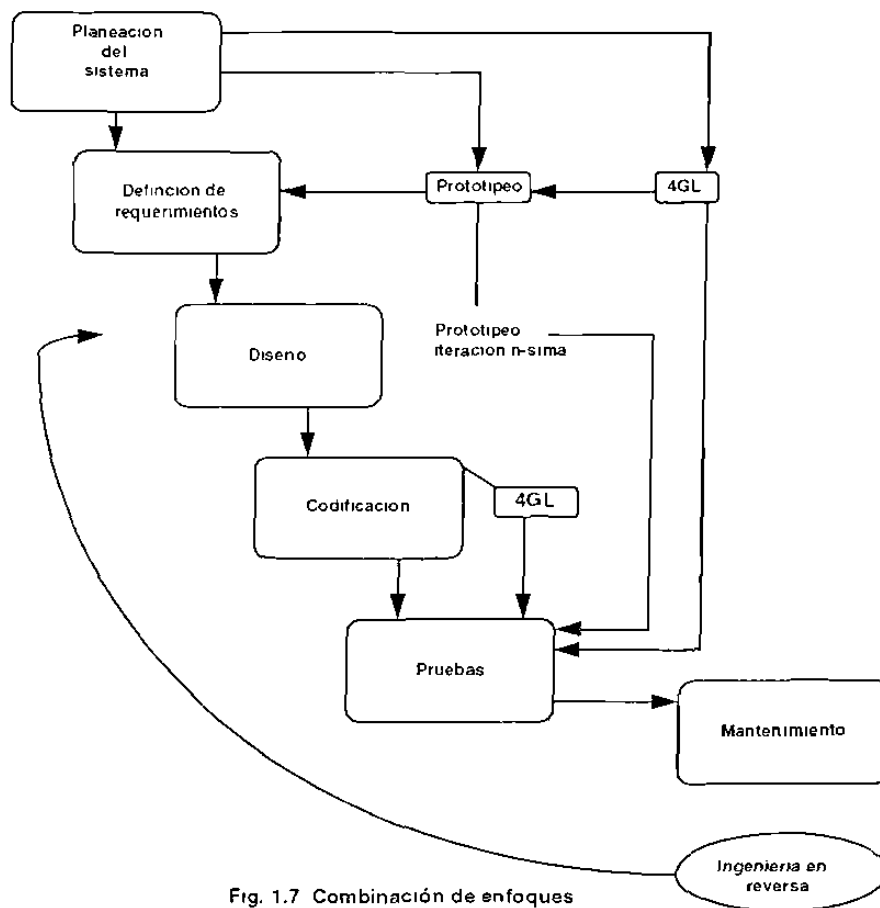


Fig. 1.7 Combinación de enfoques

En la grafica anterior podemos observar la presencia de todas las fases involucradas en el desarrollo de sistemas que se vieron en las metodologias anteriores solo que ahora estan apoyadas por tecnicas como el prototipo asi como por herramientas de cuarta generacion, que permiten disminuir el tiempo dedicado a este desarrollo y logrando asi que el desarrollador de sistemas pueda

- Realizar el desarrollo de sistemas siguiendo las fases desde la planeacion hasta la prueba del sistema para una posterior implementacion.
- Llevar a cabo las fases anteriores apoyado en herramientas de cuarta generacion para la realizacion de prototipos en apoyo a la fase de definicion de requerimientos y a la fase de construccion.
- En base a la recoleccion preliminar de requisitos construir el prototipo a probar e implementar.
- Construir el sistema tomando como base los requisitos recopilados utilizando un 4GL.

Tambien se ha incluido el concepto de ingenieria en reversa de tal forma que usted pueda mejorar o adaptar aquellos sistemas ya desarrollados y que no satisfacen ya las necesidades del usuario para lo cual sera necesario identificar sus componentes y sus interrelaciones con el fin de crear una nueva representacion del sistema

Para poder obtener un nuevo sistema necesitaremos tanto de la tecnica de ingenieria en reversa como de la ingenieria hacia adelante, que en forma conjunta forman la re-ingenieria de software, lo cual implicara llevar a cabo una investigacion y alteracion del sistema para reconstruirlo e implementarlo. Mediante el siguiente esquema podemos representar esta re-ingenieria

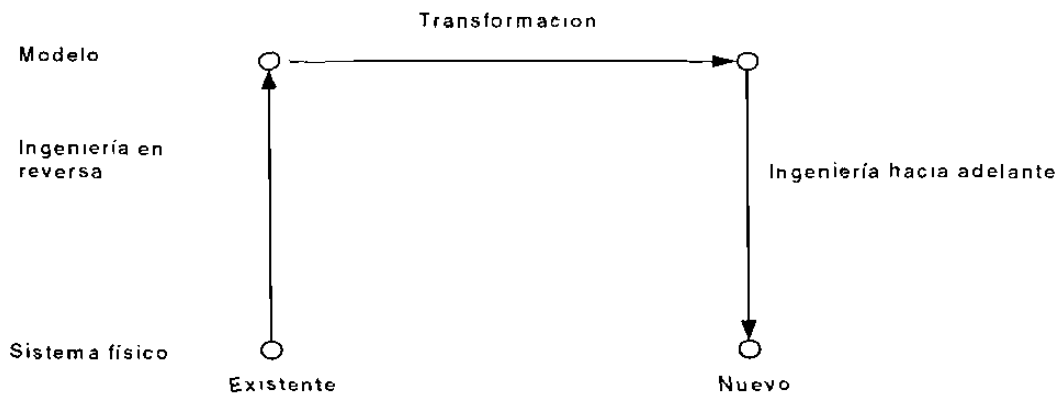


Fig. 1.8. La reingeniería en el software

Y mas a detalle mostrando los pasos comprendidos en este proceso de re-ingeniería tendríamos

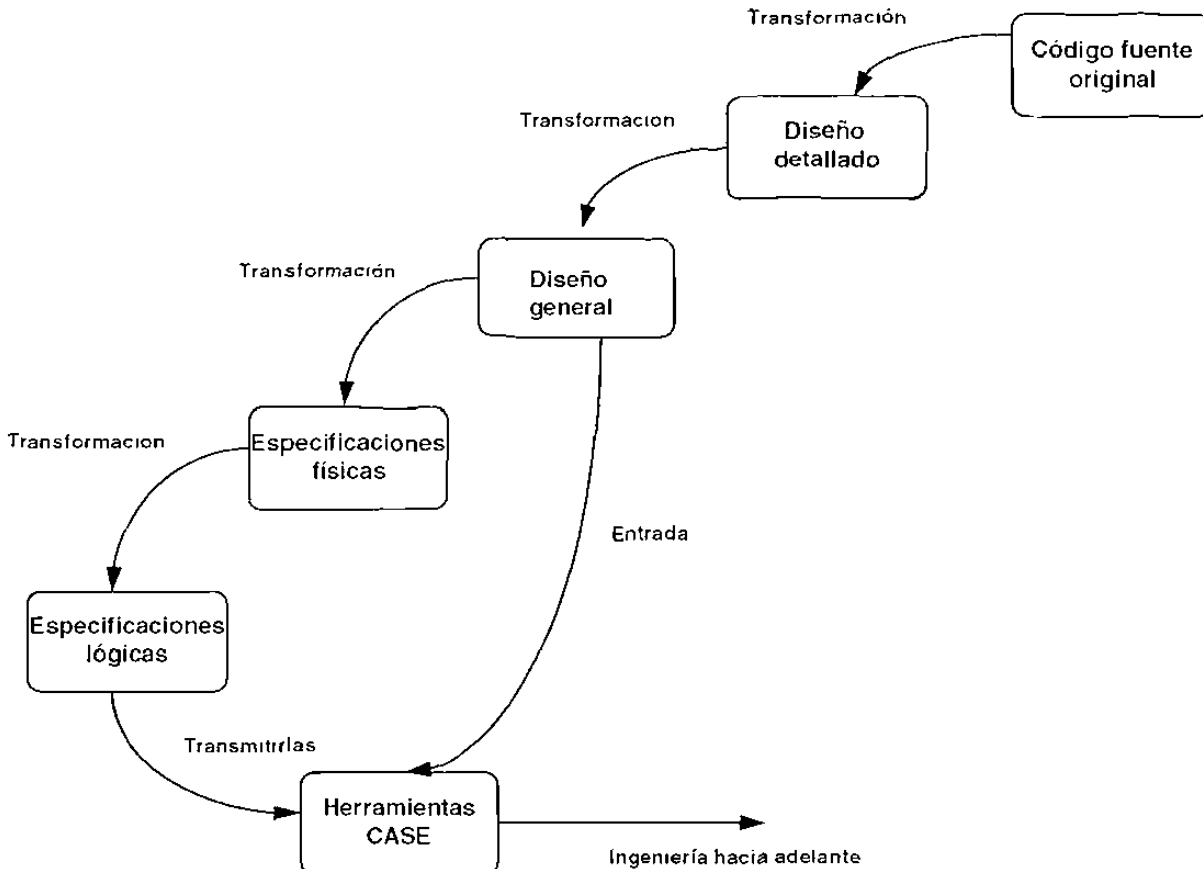


Fig. 1.9. La reingeniería y el CASE

Y si contamos con una herramienta CASE que provoca tanto la ingeniería en reversa o la re-ingeniería hacia adelante el resultado será mejor y en menos tiempo, reduciendo así los costos.

Pero hay que considerar que los problemas no tendrán una total solución aplicando solamente una re-ingeniería de sistemas o desarrollando un nuevo sistema ya que estaríamos construyendo nuevos sistemas para procesos obsoletos, lo que tendríamos que hacer para lograr tener nuevos procesos primeramente y posteriormente nuevos sistemas. lo sería aplicar la re-ingeniería de procesos. Su creador Michael Hammer coautor de 'Reengineering The Corporation: A Manifiesto For Business Revolution' la define como el acto de repensar lo fundamental y rediseñar radicalmente los procesos de los negocios y lograr así mejoras en el servicio al cliente, calidad, costos, puntualidad, competitividad, ventas e ingresos.

Para conseguir todos estos beneficios se requiere contar con una metodología que le indique el qué hacer, por ejemplo la de Thomas H. Davenport y James E. Short los cuales plantean las siguientes etapas en su metodología:

- a) Desarrollar la vision del negocio y los objetivos de los procesos
- b) Identificar los procesos a ser rediseñados

- c) Entender y medir los procesos existentes
- d) Identificar la tecnología de información
- e) Diseñar y construir un prototipo de el proceso

Aplicando así esta u otra metodología lograremos cumplir los principios tan buscados por la re-ingeniería [RUI,93]

- Pensar en grande
- Organizarse en torno a resultados.
- Hacer que quienes utilizan el producto del proceso sean los que realicen éste
- Situar el punto de decisión donde se realiza el trabajo e incorporar el control al proceso.
- Incluir el trabajo de procesamiento de la información en trabajo real que produce la información.
- Lograr procesos verdaderamente efectivos haciendo participar al cliente y al proveedor.
- Estimular al personal a través de cambios e innovaciones

Así logramos tener un proceso de calidad, apoyado con un sistema de calidad tendremos un alto nivel competitivo.

1.5. LA EVOLUCION EN EL DESARROLLO DEL SOFTWARE.

Primera era (50's).

- Desarrollo de software sin planificación.
- Solo existían especialistas
- Sistemas cuya orientación era por lotes
- Aplicaciones que procesaban grandes volúmenes de información
- Algunos sistemas eran de tiempo real.
- Software a la medida
- Aplicaciones específicas.
- El software era utilizado por la misma persona u organización.
- La persona que desarrollaba el software lo ejecutaba y lo depuraba si era necesario.
- No existía documentación
- A pesar de ser la primera era del desarrollo de sistemas se desarrollaron muchos sistemas de información excepcionales.

Segunda era (Mediados de los 60's).

- Surgen los sistemas multiusuario
- Sistemas de tiempo real con capacidad para recoger, analizar y transformar datos de múltiples fuentes
- Surgen sistemas administradores de B D

- Aparición de casas de software con el enfoque de software como producto
- Desarrollo de software para equipos mini y mainframe
- Surgen las bibliotecas de software de computadoras
- Surge el concepto de mantenimiento de software ya que los programas desarrollados *tenían* que ser corregidos cuando se detectaban fallas o por nuevos requerimientos de los usuarios
- Programas personalizados difíciles de mantener
- Inicia la administración de proyectos
- *Da inicio* la crisis del software

Tercera era (Mediados de los 70's).

- Surgimiento de las microcomputadoras
- Tecnología de software integrada en computadoras personales por expertos en hardware pero sin experiencia en desarrollo de software
- Se estabilizaba la venta de computadoras personales mientras que la venta del software crecía.
- Surgimiento de los sistemas distribuidos.
- Incorporación de la inteligencia a los sistemas
- Hardware estandar y software marcando la diferencia.
- Redes de area local y global y demanda de acceso instantáneo a la información

Cuarta era (90's).

- Desarrollo de sistemas orientados a los objetos
- Aplicaciones prácticas basadas en inteligencia artificial (sistemas expertos)
- Software con enfoque a redes neuronales
- Mantenimiento a los sistemas con el enfoque de ingeniería en reversa.
- Énfasis en las herramientas para soportar el ciclo de vida de desarrollo de sistemas
- *Computación paralela*

A pesar de todos estos cambios que han surgido de generación en generación en el desarrollo de sistemas, en cualquier organización continúa el factor de crisis del software ya que desde hace treinta años han surgido diferentes tipos de problemas como

- Metodologías de desarrollo inadecuadas así como la falta de ellas.
- No existe una visión de negocio del personal de sistemas.
- Retrasos en la entrega de sistemas
- Insatisfacción por parte del usuario
- Incremento en la asignación de recursos al mantenimiento. Originado por el desarrollo acelerado de sistemas
- Desmotivación personal (dedicados sólo al mantenimiento).
- Pérdida de oportunidades del negocio
- Sistemas deficientes
- Uso de herramientas técnicas y metodologías anticuadas
- Falta de involucramiento del usuario en las diferentes fases de desarrollo de un sistema

Esto ha dado como resultado sistemas de información que no cumplen con los requerimientos en un principio establecidos y que por lo tanto necesita entrar en una nueva fase después de su etapa de *instalación*. Además lo que se ha logrado es que los desarrolladores de sistemas inviertan mucho de su tiempo en el mantenimiento. Estudios demuestran que aproximadamente de un 50% a un 80% del costo total del software en lo que respecta a su ciclo de vida se gasta en *mantenimiento*

Pero el problema no está solamente en que no cumple con los requerimientos de los usuarios o que simplemente no funciona correctamente

El mal está también relacionado con el cómo desarrollar y cómo mantener el volumen cada vez mayor de software existente y cómo poder esperar mantenerlos al corriente de la demanda creciente de software.

Para ello los desarrolladores de software deberán afrontar y resolver esos problemas. Esto se logrará teniendo un cambio cultural en lo que respecta al desarrollo de software capacitándose en uso de metodologías, técnicas y herramientas con el fin de lograr sistemas que satisfagan los requerimientos del cliente en el tiempo y presupuesto planeados para desarrollo.

1.6. MEDIO AMBIENTE DEL DESARROLLO DE SOFTWARE.

El Medio ambiente en el área de sistemas está relacionado con las herramientas tanto de hardware y software y con las técnicas que puede utilizar un desarrollador para construir un sistema.

Hoy en día existen muchas herramientas que no sólo automatizan la fase de construcción de un sistema haciendo uso de lenguajes de programación de tercera, cuarta y quinta generación, sino que hay herramientas que apoyan las técnicas necesarias para llevar a cabo las diferentes tareas definidas en cada una de las actividades comprendidas, así como en las fases que conforman cualquier metodología de desarrollo de sistemas. De esta forma se automatiza así todo el ciclo de vida de desarrollo de un sistema permitiendo la generación automática del código, en algunos casos hasta de un 100%.

Si tenemos que desarrollar un sistema será necesario considerar que cualquier cambio en el medio ambiente de desarrollo ya sea en un lenguaje de programación, en la organización del proyecto, en el formato de las especificaciones y estándares del proyecto etc. afectará el trabajo del desarrollador, así como la cantidad de trabajo que éste pueda realizar.

Es por eso que el medio ambiente de desarrollo no sólo debe ser agradable sino flexible, para posteriores cambios y que éste a su vez nos permita medir la calidad del proceso de desarrollo de sistema y la productividad del personal encargado de este desarrollo.

Los avances en la ingeniería de software necesitan estar integrados en las metodologías de desarrollo de sistemas. Se deberá determinar realmente si una nueva técnica o herramienta mejorará el proceso de desarrollo en forma parcial o total.

'CASE' es una tecnología más de ese medio ambiente de desarrollo con el enfoque automático que soporta y facilita el desarrollo del sistema.

El acrónimo CASE que significa *ingeniería de software asistida por computadora* representa para muchos las herramientas útiles que ayudan no sólo en la fase de construcción de un sistema sino que automatizan todo el desarrollo del sistema. Su evolución en estos últimos años se representaría como

Principios 80's	Mediados 80's	Finales 80's	Principios 90's
<ul style="list-style-type: none"> - Documentación asistida por la computadora - Diagramación asistida por la computadora - Herramientas para análisis y diseño 	<ul style="list-style-type: none"> - Comprobación automática de análisis y del diseño - Depósito automático de información del sistema (repositorio) 	<ul style="list-style-type: none"> - Generación automática del código a partir de las especificaciones del diseño - Encadenamiento del diseño automático y la programación automática 	<ul style="list-style-type: none"> - Conductor de metodología inteligente - Interfaz amigable con el usuario - La reutilización como metodología de desarrollo

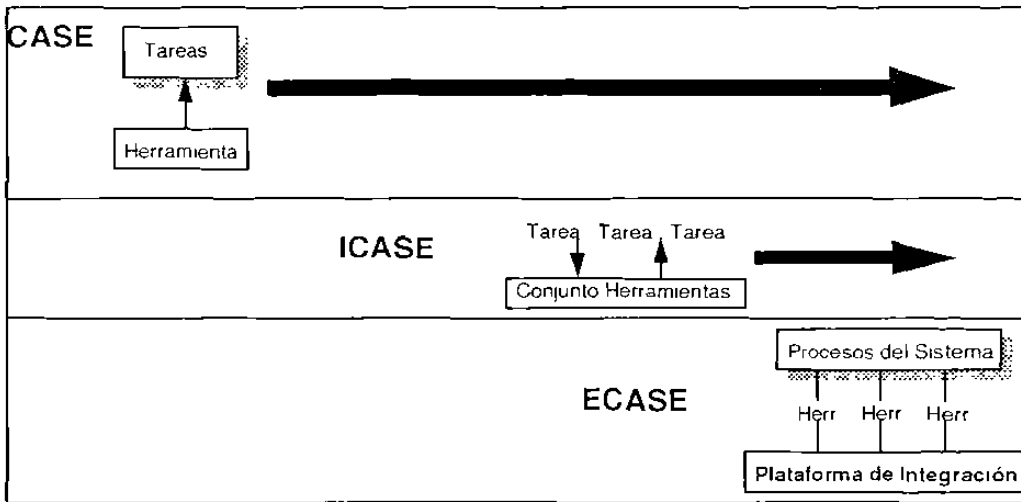


Fig. 1.10 CASE y su historia

Podríamos decir que el CASE ya no solo automatizará aquello que se hacía en forma manual utilizando *metodologías de desarrollo* como .

- Tradicional
- Waterfall
- AD/Cycle.
- Ingeniería de información
- y otras

O metodologías de análisis y diseño como:

- Jackson
- Gane & Sarson.
- Warnier-Orr.
- Ross
- DeMarco.
- Yourdon
- Top-Down.
- Bottom-up.
- Backward-Forward
- Análisis de actividades (orientado a procesos)
- Análisis Orientado a objetos.

o las tecnicas conocidas por muchos de nosotros como lo son HIPO, DFD

Quizas CASE permitira una reduccion del ciclo de desarrollo de sistemas a solamente análisis de requerimientos y especificacion de diseños, generando todo el codigo y dejandolo listo para liberarse

Tal vez esto llegue a cumplirse ya que en un futuro se pretende que incorpore componentes de sistemas expertos y que así como se logro que el producto generado por una herramienta CASE sean transportable a diferentes plataformas, CASE guiara a los desarrolladores a producir sistemas con un enfoque de calidad y por ende, incremente así la productividad

Podemos entonces afirmar que lo que para muchas personas representa CASE como sólo un conjunto de herramientas es totalmente falso ya que se considera como

El desarrollo y uso efectivo de enfoques administrativos, herramientas que automatizan y tecnicas sistematicas que permiten a los equipos de ingenieria de software producir software de acuerdo a las necesidades del negocio, disponibles en un tiempo y dentro del presupuesto permitiendo además un facil mantenimiento y realce al sistema, utilizando

Metodologias (implica seguir ciertos actividades y generar ciertos productos).

Tecnicas (que representan los caminos especificos para ejecutar los actividades de la metodologia)

Herramientas (objetos fisicos que hacen que la técnica sea mas facil de aplicar).

Pero ¿Sera CASE realmente la solucion total?, ¿Asegurará la calidad en el desarrollo de software? y ¿Se lograra una mayor productividad?

A traves de los siguientes capitulos se irán dando respuestas a estas interrogantes, conociendo primeramente el criterio de los expertos de calidad, las diferentes metodologías técnicas y herramientas que se consideran que pueden ser utilizadas para lograr una calidad en un sistema, la realidad en lo que respecta a la calidad y al desarrollo en algunas de las empresas mas importantes de nuestra localidad así como el modelo propuesto, la guía y estrategias a considerar por los desarrolladores de sistemas para lograr desarrollar e implementar un sistema de calidad.

2. CALIDAD.

2.1. DEFINICION.

Calidad para muchas personas es de más trabajo, más capacitación, más recursos no es más que aquello que se necesita hoy en día para ser competitivos a nivel nacional e internacional y para sobrevivir en cualquier arena competitiva.

El significado que le han dado a la calidad algunos autores de libros sobre este tema como Deming, Juran, Crosby, Buenrostro, Taguchi, Feigenbaum y otros es :

"Grado predecible de uniformidad y confiabilidad a bajo costo y adecuado al mercado. La calidad deberá construirse en cada fase del proceso desde la recepción de los insumos y refacciones hasta el comportamiento del producto en manos del cliente" - *Deming*.

"Adecuación al uso" - *Juran*.

"Conformidad con la especificación o cumplimiento de los requisitos" - *Crosby*.

"Profesionalismo" - *Buenrostro*.

"Mínima pérdida causada a la sociedad a partir en que el producto se embarca" - *Taguchi*.

"El producto total y las características del servicio, de mercadotecnia, ingeniería, manufactura y mantenimiento a través del cual el producto y el servicio en uso cumplirán con lo que el cliente espera" - *Feigenbaum*.

"Es desarrollar, diseñar, manufacturar y mantener un producto de calidad que sea el más económico, el más útil y siempre satisfactorio" - *Ishikawa*.

Tomando como base lo anterior diríamos que la calidad

- Debe estar orientada al usuario.
- Debe estar presente no sólo en el producto sino en todas las actividades y funciones que conllevan a la realización del producto.
- Implica no sólo cumplir con los requerimientos del cliente sino sobrepasar las necesidades y expectativas del cliente a lo largo de la vida del producto.
- Hacer las cosas bien a la primera vez.
- Cambio cultural y de sentimientos en los recursos de la organización.
- Constituye hoy en día una estrategia de la mayoría de las empresas de nuestro país y del mundo para alcanzar una ventaja competitiva, que los llevará además a lograr una *diferenciación en su producto*.
- La calidad lograda en un producto se manifestará simplemente por la satisfacción que el usuario o cliente pueda tener.

2.2. DESARROLLO HISTORICO HACIA LA CALIDAD.

A raíz de la derrota de Japón en la segunda guerra mundial, Japón se dedicó a la fabricación de productos para exportación, pero uno de sus grandes problemas era que sus productos eran de muy mala calidad.

Sin embargo gracias a que este país adoptó un modelo de administración basado en principios como:

- Hay que prevenir errores y no corregirlos.
- El trabajador debe desarrollar sus mejores capacidades, sintiéndose satisfecho.
- Hay que servir al cliente.
- Etc.

logró pasar de ser un país con productos de mala calidad, a ser un país que ofrece la garantía de calidad.

El lograr la calidad en los productos o servicios de una empresa se ha visto como en un proceso evolutivo que va desde el control de calidad mediante la inspección, hasta un control estadístico. Así se logra un involucramiento de la administración para el aseguramiento de calidad y culmina con el enfoque de la calidad como la estrategia competitiva en el momento que la alta gerencia tome como punto de partida para su planeación estratégica, los requerimientos del cliente y la calidad de los productos de los competidores.

Por otra parte los hechos más importantes que han ocurrido en torno a esa calidad buscada por muchas empresas a nivel internacional, se presentan a continuación:

<ul style="list-style-type: none"> • Control de calidad en Japón. • Nace JUSE unión nacional de científicos e ingenieros. • JUSE establece el grupo de investigación de Control de calidad ofreciendo los primeros cursos sobre esta rama • Diseño de la primera revista de control de calidad a publicarse en 1950 	<ul style="list-style-type: none"> • Primeros seminarios por Deming. • Se establece el premio Deming. • Difusión del control de calidad estadístico. • Era de la estadística con las siete herramientas estadísticas. • Sectores industriales interesados en el control de calidad 	<ul style="list-style-type: none"> • Se establece a nivel internacional el mes de la calidad. • Nacimiento de los círculos de calidad. • Se introduce el término de control total de calidad(CWQC). • Se lleva a cabo la primera conferencia internacional de control de calidad 	<ul style="list-style-type: none"> • Surgen las siete herramientas administrativas. • El QFD se pone en práctica. • Llegan a 100,000 los círculos de calidad registrados. 	<ul style="list-style-type: none"> • El control total de calidad como instrumento estratégico. • La calidad como estrategia para lograr la productividad en México. • Nace: <ul style="list-style-type: none"> -Centro de calidad del ITESM. -Asociación ex-becarios AOTS México-Japón los cuales promueven el intercambio de experiencias - Fundación mexicana para la calidad total A C - Consejo de calidad metropolitano A C. 	<ul style="list-style-type: none"> • La calidad total en la mayoría de las empresas en México. • Mayor uso de herramientas estadísticas y administrativas para la calidad
<p style="text-align: center;">Postguerra</p>	<p style="text-align: center;">50's</p>	<p style="text-align: center;">60's</p>	<p style="text-align: center;">70's</p>	<p style="text-align: center;">80's</p>	<p style="text-align: center;">90's</p>

Fig. 2.1 El desarrollo del control de calidad

2.3 FILOSOFIAS Y TECNICAS DE CALIDAD.

2.3.1. FILOSOFIAS.

Filosofía de Deming.

Edward Deming es un estadístico que ayudó a las compañías japonesas a mejorar su calidad después de la segunda guerra mundial. A él se debe en gran parte el éxito de la industria japonesa actual, ya que sugiere mejorar la calidad y la productividad así como la posición competitiva, creando constancia de propósito en la administración para la producción de bienes y servicios. Utiliza también el control estadístico para prevenir errores, mejorando el rendimiento en todas las áreas y no sólo la conformidad en cuanto a las especificaciones del producto o servicio.

Deming sugiere que para lograr establecer un programa de mejora de calidad es indispensable el compromiso de la alta dirección, siendo ésta responsable de los problemas de calidad; calidad que implica el trabajo de todos eliminando aquellas barreras que muchas veces impiden a los trabajadores a

realizar un buen trabajo y a los cuales hay que impulsarlos a trabajar más inteligentemente que duramente, a través del conocimiento de lo que se está haciendo.

No se busca conocer a través de la *costosa inspección* la cantidad de productos que son defectuosos sino *debemos buscar la garantía o mejora de la calidad*.

Es por eso que Deming recomienda 14 puntos para aquella administración que busca la calidad. Si son comprendidos y aceptados modificarán su modo de pensar y de actuar [GIT,89].

- a) Crear constancia de propósito dirigida a *mejorar* los productos y los servicios.
- b) Adoptar la nueva filosofía. No se puede seguir viviendo con los niveles de *demoras, errores, materiales defectuosos y personas no capacitadas* para la labor que desempeñan cumpliendo en forma deficiente su función sin olvidarnos también de la negligencia presente de muchos individuos.
- c) Terminar la dependencia de la inspección masiva. Inspeccionar es tardío, no efectivo y costoso ya que la calidad no viene de la inspección sino del *mejoramiento del proceso*.
- d) Acabar con la práctica de otorgar negocios a base de precios. Ya que un proveedor debe de buscar calidad y disminución de costos.
- e) *Encontrar y resolver problemas* en el sistema y mejorarlos.
- f) Instituir la capacitación en el trabajo.
- g) Instituir el liderazgo.
- h) Acabar con el miedo; el cual desaparecerá en la medida en que todos los empleados desarrollen confianza en la administración.
- i) Romper las barreras entre los departamentos.

- j) Eliminar metas numéricas, posters de lemas para la fuerza laboral. Que busquen la calidad y productividad sin estar acompañados de instrucciones o métodos de cómo hacer el trabajo.
- k) Eliminar estándares de trabajo que estipulen cantidad y no calidad.
- l) Eliminar barreras que impidan al empleado hacer un buen trabajo.
- m) Instituir un programa vigoroso de educación y capacitación.
- n) Crear una estructura en la alta administración que impulse los 13 puntos anteriores.

Filosofía de Jurán.

Se le da el crédito por el éxito japonés, siendo este país visitado por él en 1954 para dar una conferencia acerca de cómo administrar la calidad.

Creó el término adecuación al uso, con el cual indica que la adecuación al uso es juzgada por el receptor del servicio, y no es determinada por el producto.

En 1940 Jurán comentó que los aspectos técnicos del control de calidad habían sido bien cubiertos pero que las compañías *no sabían* como administrarlos para lograr la calidad identificando así problemas de organización, comunicación y coordinación de funciones en otras palabras el *elemento humano*. Jurán pensó que si comprendemos las situaciones humanas asociadas al trabajo se resolverán los problemas técnicos, tal entendimiento es un prerrequisito de una solución.

También dijo que menos del 20% de los problemas de calidad son debido a trabajadores y el resto es debido a la administración. Los administradores deben ser entrenados en calidad, involucrándolos ya que los problemas de calidad son interdepartamentales.

Además de recomendar el uso de los círculos de calidad porque permiten mejorar las comunicaciones entre la administración y los empleados, su propuesta para el mejoramiento de la calidad ha sido [JUR,90]:

- a) Crear conciencia de la necesidad y oportunidad para el mejoramiento.
- b) Establecer metas para el mejoramiento.
- c) Organizar la búsqueda de metas.
- d) Proveer *entrenamiento*.
- e) Llevar a cabo proyectos para resolver problemas.
- f) Reportar los progresos.
- g) Dar reconocimientos.
- h) Comunicar resultados(*retroalimentación*).

- i) Mantener puntuaciones.
- j) Mantener el ímpetu por hacer parte del mejoramiento anual de los sistemas y procesos regulares de la compañía.

Filosofía de Crosby.

Para Crosby la calidad no cuesta, lo que cuesta dinero son las cosas que no tienen calidad, es por eso que es importante aprender el arte de asegurar la calidad logrando que la gente (tanto la alta dirección como los niveles más bajos de la organización) tenga el hábito de realizar las cosas bien a la primera vez, *previniendo y no corrigiendo los errores.*

Para esto debe existir en la alta dirección un directivo profesional de la calidad, el cual asuma la responsabilidad de instruir a la alta dirección en la necesidad de asegurar la calidad. Asimismo, una organización que requiere y busca mejorar la calidad, la obtendrá no sólo implementando un plan sino que además debe existir un compromiso con la calidad de parte de la alta dirección.

Su programa de calidad que consta de 14 pasos está enfocado a lograr el mejoramiento de esta [CRO,90]:

- a) Compromiso de la dirección y niveles gerenciales.
- b) Formación de un equipo de mejoramiento de calidad.
- c) Medición de la calidad.
- d) Evaluación del costo de la calidad.
- e) Conciencia de la calidad.
- f) Acción correctiva.
- g) Planeación de cero defectos.
- h) Entrenamiento a los supervisores.
- i) Día de cero defectos.
- j) Fijar metas.
- k) Eliminación de las causas de error.
- l) Reconocimientos.
- m) Consejos de calidad.
- n) Hacerlo de nuevo.

Para lograr implantar este programa él indica que requiere mucha participación y conocimientos, es por eso que sugiere que el programa corporativo de calidad cuente con los siguientes apoyos:

- Activa participación de la dirección.
- Administración profesional de la calidad.
- Programas originales.
- Reconocimiento.

Además de su programa él propone cuatro principios absolutos que contribuyen a resolver todas aquellas preguntas como [CRO,91]:

- ¿Qué es la calidad?
- ¿Qué sistema se requiere para propiciar la calidad?
- ¿Qué estándar de realización deberá utilizarse?
- ¿Qué sistema de medición se requiere?

a) **Primer principio absoluto:** Calidad se define como cumplir con los requisitos.

El mejoramiento de calidad se alcanza haciendo que todo el mundo haga *las cosas bien desde la primera vez*.

Los directivos tienen tres tareas que realizar:

(1) establecer los requisitos que deben cumplir los empleados, (2) suministrar los medios necesarios para que el personal cumpla los requisitos y (3) dedicar todo su tiempo a estimular y ayudar al personal a dar cumplimiento a esos requisitos.

Crosby dice que las *personas más preparadas* y con más conocimientos se emplearán para establecer previamente los requisitos y no cuando se quiera reparar el defecto.

b) **Segundo principio absoluto:** el sistema de calidad es la prevención.

El costo más visible de la práctica de la calidad reside en el área de verificación.

La verificación ya sea que se le llame inspección, comprobación, prueba o cualquier otro nombre siempre se realiza después de que sucede algo. Si estas acciones se emplean en separar lo bueno y lo malo, lo que se hace es separar el producto bueno del malo.

La verificación es considerada como una forma clara y poco confiable de obtener calidad. Verificar, seleccionar y evaluar sólo filtra lo que ya está hecho. Lo que realmente falta es la *prevención*. Ya que el error que no existe no puede ser pasado por alto. La prevención es algo que sabemos realizar si comprendemos nuestro proceso.

La idea de la prevención se basa en el entendimiento del proceso que necesita del mantenimiento preventivo. El secreto de la prevención consiste en observar el proceso y determinar las posibles causas del error. Donde las causas pueden ser controladas.

Cada producto o servicio está formado por un gran número de componentes, los cuales deben tratarse por separado, con la finalidad de eliminar las causas de los problemas.

La solución para lograr la calidad es la prevención.

c) **Tercer principio absoluto:** el estándar de realización es cero defectos.

Una organización es un sistema con millones de pequeñas acciones, que le hacen sobrevivir y producir. Todas y cada una de estas acciones deberán realizarse conforme a lo planeado para que todo salga como debe ser.

El estándar de realización es el medio que permite a la compañía progresar.

Cuando una organización fomenta entre las personas la idea de no hacer todas las cosas bien algunas de estas acciones no se llevarán a efecto y nadie sabrá con exactitud qué se debe hacer. Para

que no existan defectos en los procesos de producción es necesario realizar todas las acciones con *exactitud*.

d) **Cuarto principio absoluto:** la medida de la calidad es el precio del incumplimiento.

El costo de la calidad se divide en dos o más: el precio del *incumplimiento* y el precio del cumplimiento. El precio del incumplimiento de los requisitos lo constituyen todos los gastos realizados en hacer las cosas mal.

El precio del cumplimiento con los requisitos es lo que hay que gastar para que las cosas resulten bien.

Filosofía de Genichi Taguchi.

El Dr. Genichi Taguchi se ha basado en el principio de que la calidad es la mínima pérdida financiera impartida a la sociedad después de que un producto es embarcado, por lo que los productos y procesos deben ser optimizados para disminuir el costo y aumentar la calidad.

Su filosofía puede darse a conocer a través de los siguientes puntos [ROY,90]:

- Cuando se evalúa la calidad de una empresa es importante considerar la pérdida total causada a la sociedad.
- Para sobrevivir en una economía competitiva son necesarios el mejoramiento continuo de calidad y la reducción de costos.
- Un programa de mejoramiento continuo de la calidad incluye la reducción incesante de la variación de las características del producto con respecto al objetivo.
- Con frecuencia la pérdida que sufre el consumidor se debe a la variación del comportamiento del producto; esta variación es aproximadamente proporcional al cuadrado de *la diferencia entre el desempeño óptimo y el desempeño actual(función de pérdida)*.

- El costo final y la calidad de un producto se determinan por *el diseño del producto y su proceso de manufactura*.
- La variación en el desempeño del producto o de un proceso se puede reducir aprovechando los efectos no lineales de los parámetros del producto o proceso sobre el desempeño.
- La planeación de experimentos estadísticos se emplea para identificar los parámetros del producto y procesos que permitan reducir la variación del desempeño.

Su función de pérdida caracteriza a todos sus métodos indicando que la pérdida se mide en términos monetarios y que ésta será sufrida tanto por el fabricante como por el consumidor. Por lo tanto, se deberá evaluar la pérdida de la calidad causada por las variaciones en el desempeño de un producto.

2.3.2. LOS EXPERTOS Y SU CALIDAD TOTAL.

A continuación se presenta el concepto de calidad total con el fin de hacer una distinción entre el criterio clásico de calidad antes definido por algunos autores contra el concepto de calidad total y la filosofía de grandes expertos de calidad.

Calidad total.

La calidad total se puede expresar como el proceso de mejora continua de la calidad de todo el sistema, de tal manera que el producto percibido por nuestros consumidores esté constantemente en correctas condiciones para su uso. La calidad total en una organización es a un tiempo: calidad de los objetivos, calidad de sistemas, calidad de información, calidad de las personas, de los trabajadores, gerentes, directivos superiores; es en suma calidad de toda la empresa u organización.

Control total de calidad.

Sistema estratégico para ofrecer en forma competitiva bienes y servicios que satisfagan los requerimientos de los consumidores. En este propósito intervienen: la alta administración, los jefes de oficina, producción, trabajo administrativo, contabilidad, materiales, almacenes, ventas, comercialización, en síntesis toda la empresa, además de todas las organizaciones relacionadas con la empresa.

Sus elementos principales son:

- El involucramiento de todas las funciones en las actividades de calidad.
- La participación de los empleados en todos los niveles en las actividades de calidad.
- El propósito de mejorar continuamente.
- Y la atención cuidadosa de la definición de calidad desde el punto de vista del consumidor.

Feigenbaum y su filosofía.

La metodología del sistema de aseguramiento de calidad da inicio con la identificación de los requisitos de la calidad del cliente y su uso final cuando el producto ya se ha colocado en las manos de un cliente que permanece satisfecho, si existe esto se puede decir que inicia la aplicación del control de calidad total entendiéndose por calidad aquello que satisface al cliente considerando el uso real del

producto y su precio de venta y entendiendo por control la realización de la revisión de: *fijación de estándares, conformidad de los estándares, el tomar acciones cuando se excedan los estándares y la planeación de mejoras a los estándares.*

Aplicar el control total de calidad implicará [FEI,91]:

- Controlar los nuevos diseños.
- Controlar los materiales adquiridos.
- Controlar el producto.
- Llevar a cabo estudios especiales del proceso.
- Y tener un control sobre los sistemas tecnológicos y humanos.

Ishikawa.

En su libro *¿Qué es control total de calidad?* nos indica que el control de calidad implica que en la empresa *todos promuevan y participen en el control de calidad* [ISH,86].

Esto significa que la gente involucrada en todos los niveles organizacionales (desde la alta administración hasta el nivel operativo) debe seguir la filosofía de control de calidad.

Aplicar el control de calidad consiste en:

- a) Conocer los *requisitos del consumidor* de tal manera que se establezcan las características *reales del producto*.
- b) Establecer las características sustitutas.
- c) Entender la relación entre las características sustitutas y las reales.

Siendo la calidad real lo que el cliente solicita y la calidad sustituta lo que el productor establece para lograr la calidad.

Determinado entonces estas características de calidad reales es importante seleccionar el lenguaje que utilizaremos para expresarlas.

A continuación se muestran algunas ideas de cómo expresarlas:

- Determinar la unidad de garantía.
Se debe conocer de que manera se va a medir la unidad de garantía.
- Determinar el método de medición.
Conocer de que manera se va a medir el producto, definido previamente por el departamento de producción.

- Determinar la importancia relativa de las características de calidad.
Ya que el producto tiene muchas características de calidad.
- Llegar a un consenso sobre defectos y fallas.
Las opiniones en cuanto a defectos y fallas varían según los fabricantes, consumidores y el personal de la empresa.
Siendo los defectos los factores que no están correctos pero que no afectan al funcionamiento, y las fallas sí afectan al funcionamiento.
- Revelar los defectos latentes.
Para muchas personas los productos defectuosos son aquellos que se deben rechazar y evitar.
Sin embargo, existen productos que tienen defectos ocultos o que no se ven a simple vista.
- Observar la calidad estadísticamente.
Debido a que no todos los productos son iguales con un programa estadístico se puede observar claramente dónde están las fallas y cuáles son las causas.
- Calidad del diseño y calidad de aceptación.
La calidad del diseño es la forma como está diseñado el producto. Y la *calidad de aceptación* es la manera en que los productos se ciñan a la calidad del diseño.
Si existiera alguna diferencia entre los dos tipos de calidad el costo sube debido a los defectos.
Sin embargo si la calidad de aceptación es mayor, el costo baja.

Ahora bien para realizar el control de calidad es importante seguir los siguientes pasos:

a) Determinar *metas y objetivos*.

Si no se fijan las políticas no se pueden establecer las metas. La determinación de las políticas corresponde a la alta gerencia.

Determinada una política, las metas se hacen evidentes por sí mismas. Deben expresarse concretamente en cifras y con un propósito. Evitar órdenes abstractas como " estudien " o controlen eficazmente.

Estas metas deben resolver los problemas de la empresa, deberán estar por escrito, distribuir las hasta el nivel más bajo en el organigrama y deben de tener un plazo claro de tiempo.

b) Determinar métodos para alcanzar las metas; normalización de trabajo.

La determinación de un método equivale a normalización. Esto significa que si una persona desarrolla un *método*, deberá normalizarlo, convertirlo en reglamento y luego *incorporarlo dentro de la tecnología y propiedad de la empresa*.

Este método deberá ser útil para todos y libre de dificultades.

c) Dar *educación y capacitación*.

Los superiores tienen la responsabilidad de capacitar a sus subordinados.

Cuando el subordinado ya capacitado esté listo, se le delega autoridad y se le da la libertad de hacer su trabajo. Al tener la capacitación, el empleado será digno de confianza y no requerirá de supervisión excesiva.

d) Realizar el trabajo.

Si se realizan los pasos anteriores no se debe tener ningún problema al ejecutar el trabajo.

e) *Verificar* los efectos de la realización.

Si verificamos esto y aquello no se alcanzará el éxito pues esta forma de gerencia se basa en el supuesto que el hombre es malo por naturaleza.

Lo más importante en la gerencia es el principio de excepción. Si las cosas se desarrollan de acuerdo con las metas y normas fijadas, entonces se deben dejar que sigan así. Pero si surgen hechos inesperados o situaciones no rutinarias, el gerente debe intervenir. El objeto de verificar es descubrir tales excepciones. Para cumplir esta tarea es necesario entender con claridad las políticas básicas, las metas y los procedimientos de normalización y de educación.

f) Tomar la acción apropiada.

Es importante tener medidas para impedir que las excepciones vuelvan a repetirse.

Es necesario detener las irregularidades. Al eliminar las causas de las excepciones hay que remontarse al origen del problema y tomar medidas para evitar que se repitan.

Buenrostro.

Su filosofía se enfoca principalmente en que la calidad se logra [BUE,90]:

- Trabajando en equipos siendo estos sobresalientes.
- Reconociendo que la organización es todo un equipo complejo.
- La organización debe tener una capacidad del proceso la cual depende de las variabilidades de insumos y de sus procesos.
- Todos los miembros de la organización deben atender sus problemas en forma prioritaria, corregir las causas desde su raíz y establecer procedimientos que minimicen su repetición.
- Cambiar la forma de actuar utilizando las herramientas adecuadas para que la empresa construya su cultura de calidad.
- Uso del control estadístico de procesos. Implementar en etapas.
- Clarificación de roles. Evitar la ambigüedad de roles.
- Utilizar una metodología adecuada para la implantación de catálogos de productos y servicios como:
 - a) El primer nivel de la organización acepta el enfoque y los principios de la calidad que regirán el desempeño de todos en la empresa.
 - Preparación intensiva al primer nivel.
 - Promulgación de los principios de la calidad.
 - Difusión de los principios de la calidad.
 - b) Sensibilización hacia el proyecto y capacitación en la técnica de catálogos.
 - c) Elaboración de los catálogos de productos y servicios:

- Contribuciones.
 - Roles.
 - ¿Quién es el cliente?
 - Indicadores.
 - Colaboraciones requeridas .
- d) Negociaciones cliente-proveedor.
- e) La semana "Cero Excusas".
- Planeación.
 - Cuantificación del status inicial.
 - Arranque.
 - Evaluación y clausura.
- f) Monitoreo(semanal, mensual, trimestral).
- g) Diseñar el sistema formal de desempeño.
- h) Elaboración de los manuales de "operación estándar".
- i) Repetición del Procedimiento.
- j) La etapa del mejoramiento continuo.

2.3.3. TECNICAS.

Círculos de calidad.

Esta importante técnica para el control total de calidad existe en Japón desde 1962 siendo hoy importante en muchas empresas de más de 50 países.

Los círculos de calidad son :

Grupos pequeños formados por empleados que realizan su tarea en un área común y que se reúnen voluntaria y periódicamente con su encargado o animador para identificar las causas de los problemas en sus trabajos, proponer soluciones a la gerencia y en dado caso implantarlas.

La idea básica de las actividades realizadas por estos círculos de calidad es:

- Contribuir a la mejora de desarrollo de la empresa.
- Respetar a la persona y construir un ambiente tranquilo donde trabajar tenga sentido.
- Aprovechar completamente las capacidades humanas y desarrollar las infinitas oportunidades que se les ofrecen.

Su éxito depende de que la empresa haya aceptado la nueva filosofía de calidad. Y de que la alta dirección los apoye asumiendo responsabilidades de aceptar la participación de los empleados; proporcionar la infraestructura de apoyo, seleccionar y preparar a un responsable que apoye e instruya a los círculos y crear un sistema que reconozca sus logros.

Las bases fundamentales de sus actividades lo constituirán:

- a) Autodesarrollo.
- b) Voluntariedad.
- c) Actividad en grupo.
- d) Participación de todos.
- e) Utilización de las técnicas de control de calidad.
- f) Robustecimiento y continuidad de las actividades de los círculos de calidad.
- g) Desarrollo mutuo.
- h) Creatividad.
- i) Conciencia de la calidad.
- j) Conciencia de problemas.

Ya que su alcance será limitado, se resolverán problemas muy específicos y aunque su formación sea estimulada e inducida por otras actividades del plan maestro, los miembros del círculo de calidad asistirán a sus reuniones (fuera de sus horas de trabajo) hasta la conclusión del proyecto.

Cullen y Hollingum afirman que el gran valor de los círculos de calidad reside en la educación, motivación y comunicación. Esto logra que los miembros del equipo estén motivados para desempeñar su trabajo en el logro de los objetivos de calidad.

Grupos de calidad.

Este tipo de grupos también ayudan a la solución de problemas, el cual está integrado por empleados orientados a resolver problemas o a establecer mejoras de carácter administrativo, pueden participar personas que no necesariamente pertenecen a un área específica.

Equipos de mejora.

El equipo de mejora es un grupo formado por personas de la administración media, que pertenecen a diferentes áreas. Esto se hace con el propósito de resolver problemas o desarrollar una mejora conforme al programa anual de mejoramiento y reducción de costos, y el de control y mejora de funciones administrativas.

Podemos afirmar que estos equipos de mejora resuelven aspectos de vital importancia estratégica, mientras que los círculos de calidad sólo solucionan problemas operativos en su lugar de trabajo. Pero tanto, los equipos de mejora, grupos de calidad y círculos de calidad, logran satisfacer las necesidades del ser humano en cuanto a autorrealización, reconocimiento y seguridad se refiere.

Brainstorming y el diagrama de seis palabras.

Esta técnica conocida como lluvia de ideas estimula la creatividad, poniendo en marcha el flujo de ideas y proporcionando confianza a un grupo de personas. Elimina todas las inhibiciones que impiden el flujo de ideas, tales como el deseo de no comprometerse, el temor a recibir críticas de otras personas, la falta de confianza en el propio juicio, etc.

Las fases de una sesión de lluvia de ideas son:

- Presentar el tema e indicar el límite de los mismos.
- Cada participante, por turno, da una idea que es registrada a la vista de todos por un anotador.
- Y agotada la fase productiva se realiza la selección.

Diagrama de seis palabras.

Técnica de análisis sencilla y de gran utilidad para asegurar que se pregunta lo más importante de cada situación o problema que se trate de analizar.

Las seis palabras son: Cuándo, Cómo, Dónde, Qué, Quién y Por qué.

Primeramente, se describirá el problema, para dar lugar a dibujar en una tabla todas estas palabras separadas por leyendas de "si es problema", "no es problema".

Posteriormente, se deberá aplicar el *brainstorming* para cada una de las palabras, una vez terminado con todas deberá dedicarse un tiempo extra para que otros den su opinión.

Si hubiera numerosas causas, el grupo debe preguntarse por el 20% de las causas que crean el 80% del problema.

Las causas explicarán tanto cuándo es problema, como cuándo no es problema.

2.4. CALIDAD Y PRODUCTIVIDAD.

Productividad cuyo significado se relaciona con obtener más de lo que se invierte, esta fuertemente ligado con la calidad.

Realizar las cosas bien a la *primera* vez que se ejecuten constituye ese vínculo entre la calidad y la productividad. Ambos conceptos relacionados representan la eficiencia en la conversión de los insumos, con un mismo objetivo "obtener más productos utilizables con el mismo gasto de insumos", el hecho es que no hay calidad sin productividad, esta productividad no tendrá sentido si no se incrementa o se mantiene la calidad.

Para ello sería necesario eliminar correcciones, revisiones o inspecciones y retrabajos lo cual implicaría un cambio continuo y permanente con un enfoque de calidad basado en un proceso de cambio cultural en la organización con un enfoque a educar, con un liderazgo de la dirección general, con una total administración del proceso de calidad donde cada uno supervise y verifique su propio trabajo, logrando así ser más competitivos en la empresa.

Mejorando la calidad, mejorará la productividad logrando acrecentar la satisfacción del cliente.

- Esto significa que si logramos alta calidad y baja productividad se obtendrán productos excelentes a un alto costo.
- Con alta productividad y baja calidad se obtendrán productos chatarra.
- Y haciendo bien las cosas a la primera obtendremos:

CALIDAD + EFICIENCIA = PRODUCTIVIDAD.

2.5. EN BUSCA DE LA CALIDAD: CAPACITACION.

El cambio cultural indispensable para lograr la calidad debe darse en las personas que conforman a la organización. Este cambio estará dado por un programa de capacitación que tendrá como objetivo enfocar las creencias, valores, normas e ideologías de la persona a la calidad.

Si la capacitación es considerada como un medio para lograr la calidad, permitirá que la persona se supere constantemente con el fin de ser o hacer mejor las cosas. Para ello necesitará tener una actitud positiva, conocimientos sobre el tema, habilidades y, cumplir con un programa de calidad.

Se recomienda realizar un diagnóstico para detectar las necesidades de capacitación con el fin de preparar un programa adecuado.

Preparar a las personas no es "malgastar el dinero" representa una inversión hacia el camino de la calidad.

Muchos de los puntos de vista de nuestros expertos en calidad, así como las técnicas y herramientas por muchos utilizadas y probadas para el control de calidad nos ayudarán a tener una mayor visión sobre cómo podremos lograr y controlar la calidad en un CVDS, si logramos aplicarlas en forma efectiva.

Esperamos que la información de los siguientes dos capítulos ayude a conocer un poco más sobre aquellas metodologías, técnicas y herramientas con enfoque a sistemas que junto con lo expuesto por los autores de calidad nos ayude a ir formando ese modelo, guía y estrategias para que se logre producir un sistema con calidad.

3. METODOLOGIAS, TECNICAS Y HERRAMIENTAS QUE MEJOR SOPORTAN EL DESARROLLO DE SISTEMAS.

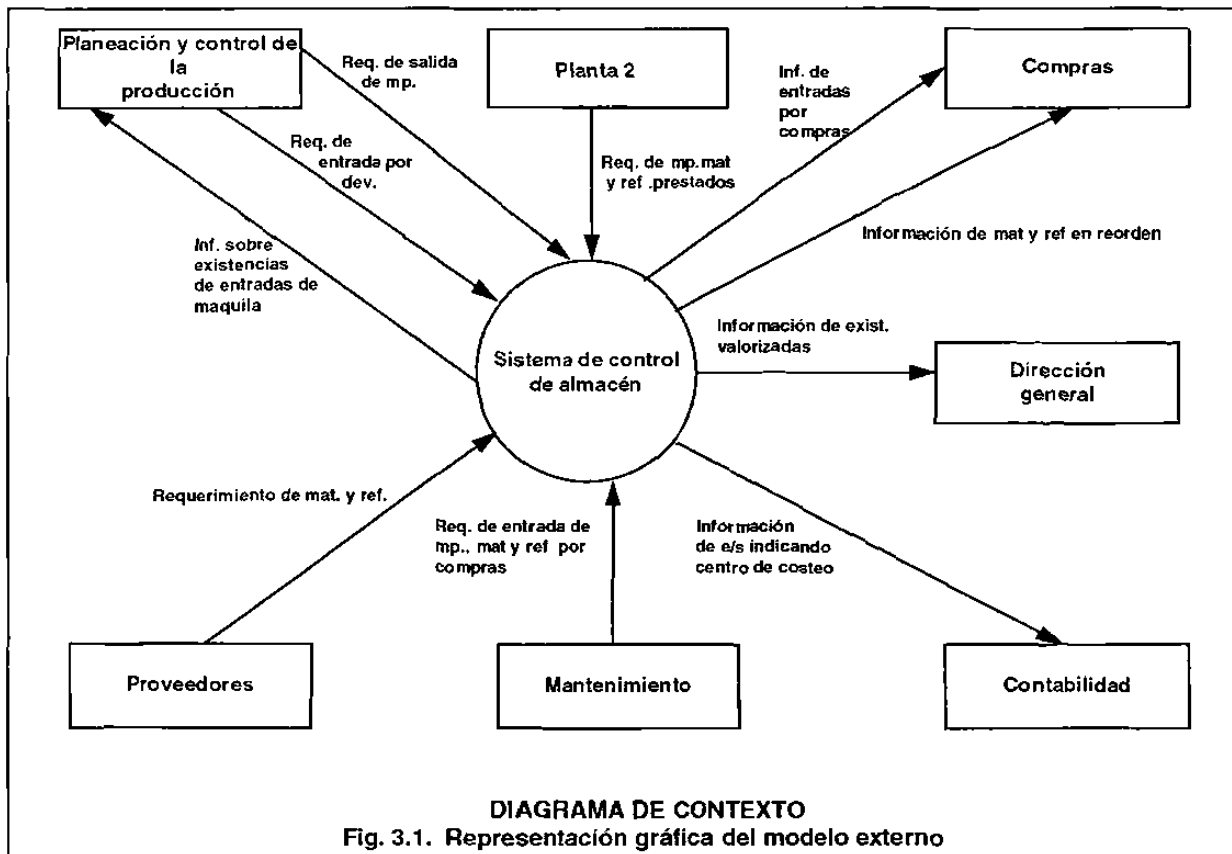
3.1. ANALISIS Y DISEÑO ESTRUCTURADO.

Esta técnica que soporta las fases de análisis y diseño en el desarrollo de sistemas basados en una metodología estructurada, ha sido desarrollada por diferentes autores como : De Marco, Gene & Sarson y Yourdon.

Haciendo mención a esta técnica no pretendemos explicar en forma detallada los diferentes pasos en el proceso de análisis y diseño estructurado, sino sólo recordar al lector de este proyecto de su existencia y su uso en la actualidad en el desarrollo de sistemas con herramientas que la soportan como CASE.

De manera general, podríamos hablar de la técnica de Yourdon por el hecho de que su técnica ha sido la que más se ha actualizado, siendo la última "la técnica de análisis estructurado moderno". Los productos que esta técnica genera son: el modelo externo e interno.

Siendo el modelo externo aquél que nos indica la función y la ubicación del sistema propuesto en relación a su medio ambiente, describe la relación que el sistema tiene con diferentes elementos externos que proporcionan o que reciben información del mismo.



El diagrama de contexto se considera una parte importante del modelo externo ya que delimita la función del sistema por la información que acepta y genera, mostrando quiénes interactúan con el sistema, que eventos son entradas y a los cuales el sistema deberá de responder así como los eventos que pueden generarse fuera del sistema.

Una vez que se tiene este modelo externo, se procede a la construcción del modelo interno donde se describen los diferentes procesos que se realizan en el sistema, así como el flujo de información entre ellos, el comportamiento de ese flujo de información y la información que se almacena.

Para ello se requiere utilizar la herramienta de diagrama de flujo de datos (DFD) la cual muestra gráficamente los procesos conectados entre sí mediante el flujo de datos.

Cada proceso si se desea puede ser dividido mediante otra red de procesos logrando así diagramas de flujo de datos refinados.

Deberá balancearse y checarsé la consistencia en estos DFD y definir las miniespecificaciones, modelo, entidad, relación, diccionario de datos y diagramas de transición de estados.

Por otra parte el diseño estructurado el cual necesita del modelo conceptual del sistema, generará un modelo conceptual computacional cuyo objetivo es mostrar el diseño de interfases, de programas y de datos, soportado por los diagramas estructurados. Mostrará así no sólo la estructura general del sistema sino que permitirá descomponer en módulos de diferentes niveles, presentando además los diferentes módulos que componen al sistema el flujo de información de datos y de control en las llamadas de un módulo a otro.

Esta técnica en forma manual es difícil de aprender, más aún si son soportadas por las herramientas automatizadas. El hecho es que si contamos con un software que permita la creación de estos modelos conceptual y computacional, lograremos grandes beneficios.

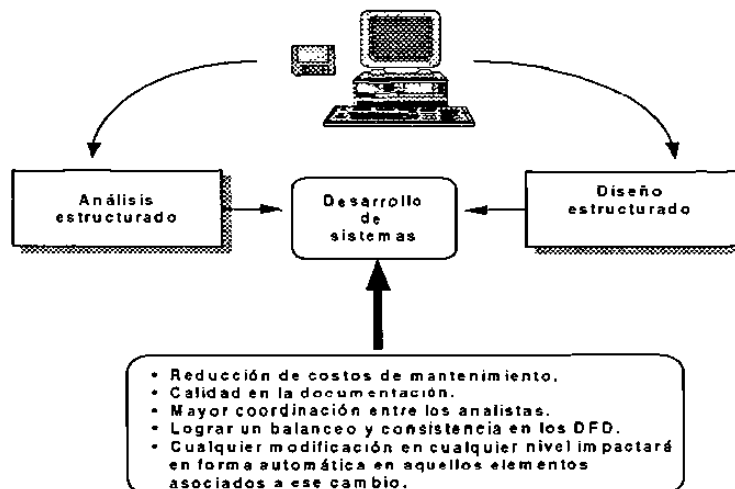


Fig. 3.2 Beneficios con las herramientas CASE que soportan Análisis y diseño estructurado

3.2. INGENIERIA DE SOFTWARE.

La ingeniería de software es una disciplina tanto administrativa como tecnológica que establece y utiliza los principios de la ingeniería (métodos) para la producción y mantenimiento del software, cumpliendo las restricciones de tiempo y costo. Comprende tanto los métodos, procedimientos y herramientas permitiendo que el administrador del proyecto tenga un control sobre todo el proceso de desarrollo, proporcionando además al equipo de desarrollo de sistemas las bases para construir un software de calidad.

Su enfoque está centrado en la integración entre la planeación, administración y desarrollo los cuales se intercalan para dar como resultado un sistema de información. Los beneficios que se obtienen al aplicarlo son:

- La reducción del mantenimiento del software.
- Incrementar la satisfacción del usuario.
- Terminación de los proyectos en el tiempo previsto y dentro del presupuesto.
- Incrementar la habilidad para planear y calendarizar los proyectos de desarrollo futuros y optimizar la asignación de recursos.
- Sistema de fácil operación.
- Sistema adaptable a un medio ambiente dinámico.

Logrando principalmente mejorar la calidad del software, incrementando la productividad, abatiendo así la crisis del software por los beneficios que se obtienen de ella.

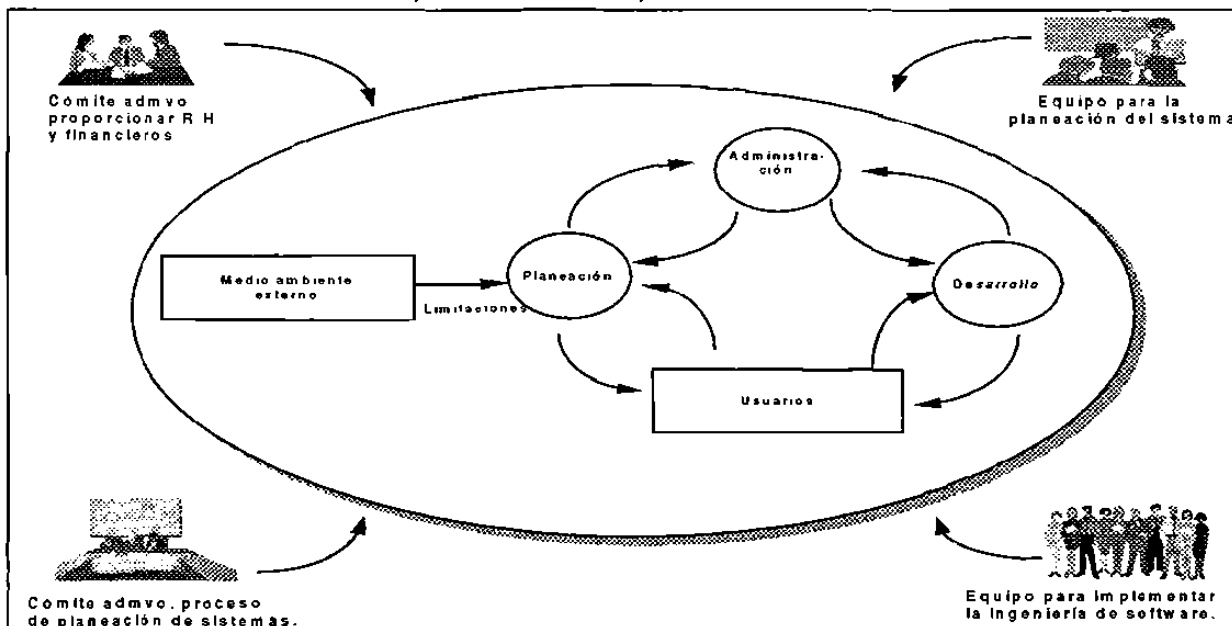


Fig. 3.3 La ingeniería de software en el desarrollo de sistemas.

Su implementación requerirá de todo un plan llevando a cabo las siguientes fases:

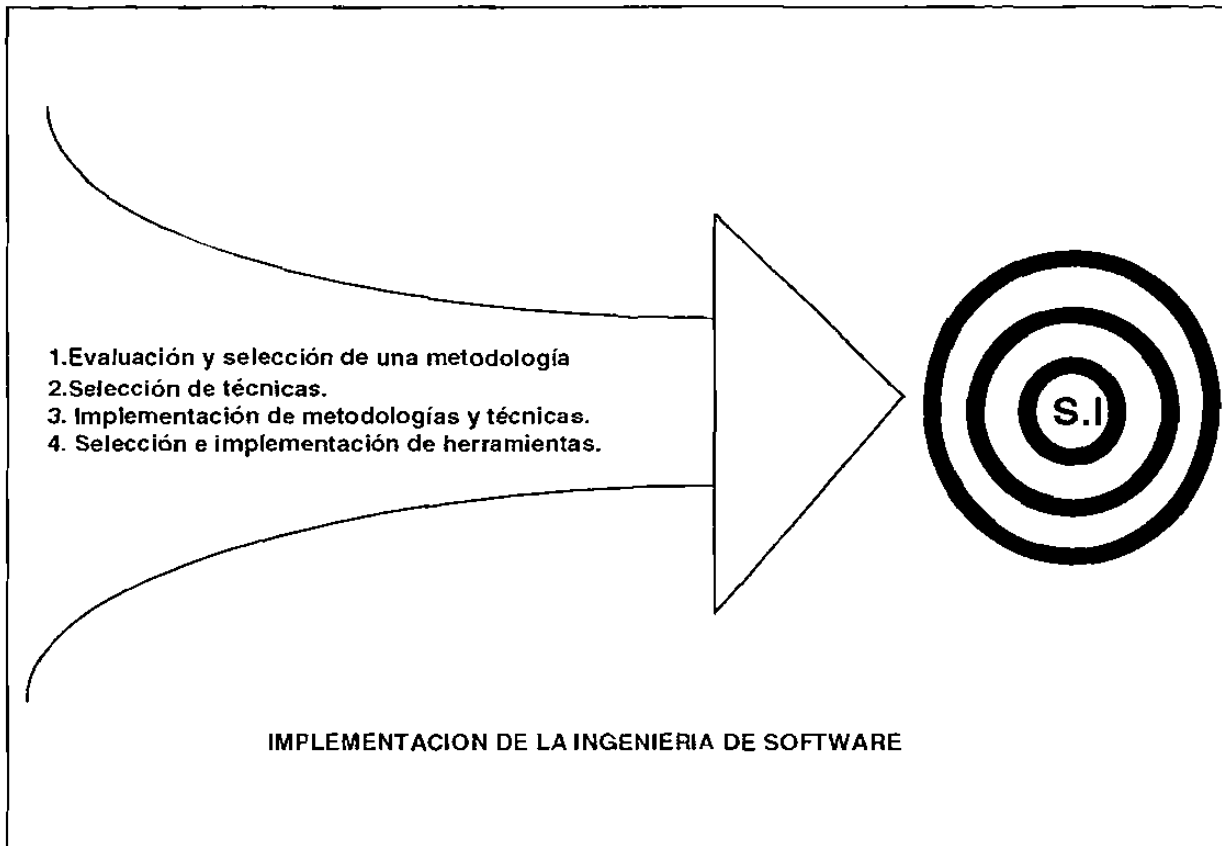


Fig.3.4 Actividades a seguir para la implementación de la ingeniería de software.

La integración se considera como la fase final, ya que las metodologías técnicas y herramientas evaluadas deben encajar. La integración es también un verdadero proceso de cómo nuevas tecnologías llegan a estar disponibles, por eso la ingeniería de software no es un proceso estático, debe ser flexible para permitir que nuevas técnicas y procesos sean fácilmente integrados.

Resulta muy importante buscar la integración entre las metodologías, técnicas y herramientas. Más aún si la ingeniería de software es un proceso flexible, éstas podrán integrarse después de una evaluación para mejorar la eficiencia y la calidad en el proceso del desarrollo de sistemas de información.

3.3. INGENIERIA DE INFORMACION.

La ingeniería de información consiste en aplicar un conjunto de técnicas formales para llevar a cabo la planeación, análisis, diseño y construcción de sistemas de información en toda una empresa o en uno de los mayores sectores de la misma.

Es considerada como una metodología orientada a los datos y no a los procesos, por el hecho de que estos últimos tienden a cambiar frecuentemente; esta metodología trata de establecer una estructura no-cambiante de datos sobre la cual será posible realizar el desarrollo de sistemas.

A través de las técnicas que se utilizan se logran romper las barreras de comunicación entre los usuarios, la administración en sus diferentes niveles organizacionales y el personal del departamento de informática. Esto permite lograr un análisis profundo de los datos usados para hacer funcionar a la empresa, y alimentar esos datos en sistemas que permitan nuevos usos o nuevas formas de tomar decisiones.

Las cuatro etapas que componen a la ingeniería de información son [MAR,89]:

- a) **Planeación estratégica de información.**
Interesada en las metas de la alta administración y los factores críticos de éxito. Interesada con el cómo la tecnología puede ser utilizada para crear nuevas oportunidades o ventajas competitivas.
- b) **Análisis de áreas de negocio.**
Determinar qué procesos son necesarios para un área de negocios seleccionada, cómo se interrelacionan y qué datos se necesitan.
- c) **Diseño de sistemas.**
Se interesa en el cómo los procesos seleccionados en el área de negocios son implementados en procedimientos y cómo estos trabajan. El involucramiento del usuario final es necesario en el diseño de los procedimientos.
- e) **Construcción.** Implementación de los procesos utilizando generadores de código, lenguajes de cuarta generación y algunas herramientas para usuario final.

Las metodologías que provee la ingeniería de información y que soportan a estas etapas se representa como bloques donde cada uno de ellos es dependiente de los que se encuentran debajo de él.

Bloque 1: Planeación estratégica de información y el modelo de la empresa.

Deben determinarse los objetivos de la empresa así como la información que se requiere para lograrlos.

Bloque 2: Análisis de entidades. Es un análisis tipo top-down de los tipos de datos que deben conservarse y cómo ellos se relacionan con otros.

Este análisis podrá realizarse a toda la empresa, a cada división, subsidiaria o porción de la empresa.

Bloque 3: Modelación de datos. El análisis de información examina los tipos de datos necesarios. Se crea un modelo de información, el cual es una visión general pero no contiene todos los detalles para la implementación de la B.D.

La modelación de datos crea el diseño detallado de la B.D. lógica y trata de hacerla tan estable como sea posible antes de que sea implementada.

Es una extensión de el bloque 2, normalizando los datos y aplicando diversos chequeos de estabilidad.

Bloque 4 y 5: Muchos lenguajes y generadores son utilizados por los usuarios, logrando así que los usuarios no inventen sus propias estructuras de datos. El uso de los lenguajes de usuario final debe estar ligado a los modelos de datos representados por estos bloques.

Bloque 6: Se refiere al análisis de una específica área del negocio. Las funciones de esa área son descompuestas en procesos utilizando diagramas de descomposición. Diagramas de dependencia o diagramas de flujo de datos se pueden utilizar mostrando así la interrelación de los procesos.

También debe incluirse una matriz que muestre cuáles procesos usan, cuáles entidades y cómo las utilizan.

Bloque 7: Diseño de procedimientos: Este utiliza múltiples tipos de diagramas, para representar aquellos eventos que cambian o usan la B.D. y agruparlos dentro de los diagramas de procedimientos.

Estos diagramas de procedimientos pueden posteriormente convertirse a especificaciones en español estructurado.

Bloque 8 y 9: El diseño creado en el bloque 7 puede ser directamente implementado con un generador de códigos o un lenguaje de cuarta generación, inclusive se puede crear un prototipo, el cual se puede modificar hasta llegar a ser el código operacional.

Bloque 10 y 11: El prototipo quizás requiere convertirse a otro lenguaje logrando un mejor desempeño de la computadora.

Bloque 12: Análisis de uso de datos da un medio formal para recolectar y diagramar el uso de la información listo para el diseño físico de la base de datos. Aquí se decidirá cuales serán las secuencias de búsquedas más frecuentes, sus volúmenes y necesidades de tiempos de respuesta.

Bloque 13: Existen muchas razones para distribuir sistemas y muchas formas de distribución. Los modelos de datos pueden ser o no divididos para su implementación, en bases de datos separadas.

Bloque 14: Convertir los modelos de datos y procedimientos en el diseño físico de la base de datos.

Bloque 15: La implementación puede ser hecha con generadores de código o lenguajes de cuarta generación.

Bloque 16: Cuando el sistema está operando, el mantenimiento podrá realizarse si se necesita, cambiando diagramas y regenerando cualquiera que sea posible.

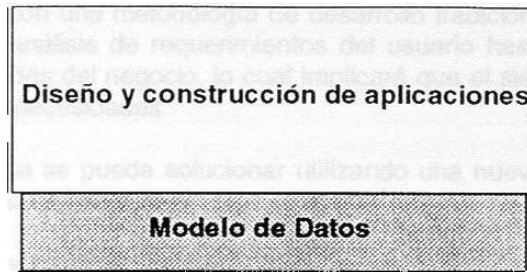


Fig.3.5 Lo que representa la ingeniería de información

Considerando así las etapas y metodologías de la ingeniería de información se podrá:

- Identificar las oportunidades de sistemas estratégicos y lograr una ventaja competitiva para construir este tipo de sistemas antes de la competencia.
- Enfocar el procesamiento de datos a las metas del negocio.
- Sistemas diferentes estarán coordinados.
- Administrar la información de tal manera, que los tomadores de decisiones clave pueden tener la mejor información disponible.
- Nuevos sistemas podrán construirse relativamente en forma más rápida, utilizando poderosas herramientas dentro de una estructura de ingeniería de información.
- Cambiar los procedimientos computarizados rápidamente.
- Tener grandes ahorros a través del uso de diseño y código reusable.
- Reducir drásticamente los problemas de mantenimiento y de trabajo atrasado.
- Computarizar toda la empresa.

Gracias a todos estos beneficios ya no se tendrán más sistemas ni información aislada, satisfaciendo así los requerimientos de cualquier usuario en la empresa.

3.4. RAD.

El problema que existe con una metodología de desarrollo tradicional de sistemas es que en el tiempo que transcurre entre el análisis de requerimientos del usuario hasta la operación del sistema, existen cambios en las necesidades del negocio, lo cual implicará que el sistema recientemente liberado ya no cumpla con estas nuevas necesidades.

Hoy en día este problema se puede solucionar utilizando una nueva metodología que pretende desarrollar sistemas en una forma rápida a bajo costo y con alta calidad. (RAD).

Esta metodología RAD sugiere utilizar [MAR,91]:

- a) Equipos pequeños.
- b) Partes reusables.
- c) Herramientas automatizadas.

Además involucrando al usuario en todo el ciclo de vida de desarrollo del sistema se logrará encontrar los requerimientos del negocio o del usuario tan efectivamente como sea posible.

Esta nueva metodología se compone de cuatro fases:

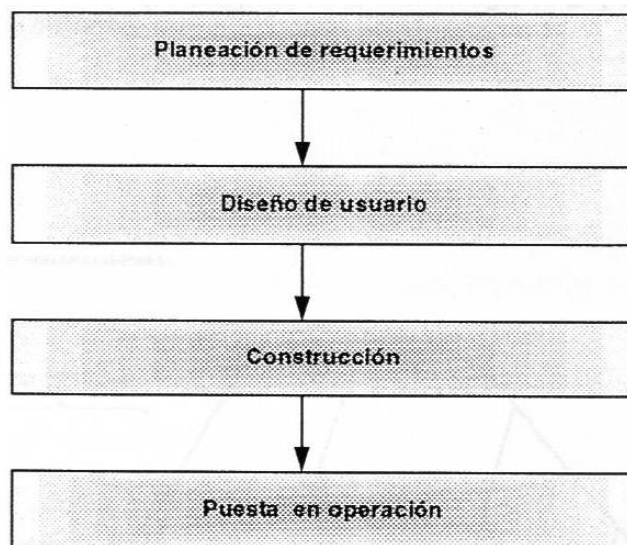


Fig. 3.6 Etapas del desarrollo rápido de aplicaciones

- a) La fase de planeación de requerimientos.
Requiere que los usuarios de alto nivel determinen qué funciones del sistema deben considerarse y qué problemas son necesarios resolverse.
- b) La fase de diseño de usuario.
Se requiere que los usuarios participen en un diseño no técnico del sistema guiados por los profesionales de Sistemas de Información.
Es aquí donde se puede hacer uso de técnicas y herramientas como JAD, prototipo e I-CASE.
El resultado de esta fase ya no será un simple diseño en papel sino un diseño representado en una herramienta I-CASE, la cual será tomada directamente por la fase de construcción.
- c) Fase de construcción.
Tomando el diseño creado en la fase anterior y usando un generador de código, se podrá generar un código libre de errores que puede corregirse inmediatamente. Sólo se necesitará de un analista para la realización del diseño y de la programación.
- d) Fase de puesta en operación.
Cuando esto sucede una variedad de acciones se requieren: probar el sistema, entrenar a los usuarios, cambios organizacionales y operación en paralelo con el sistema previo hasta que el nuevo sistema se instale.

Es importante hacer hincapié que al comienzo de cada una de estas fases, debe haber aprobación por parte del usuario ejecutivo, de quien aprueba el presupuesto.

Si consideramos las fases que componen a la metodología RAD y realizamos una comparación gráficamente el tiempo de desarrollo se vería enormemente reducido a:

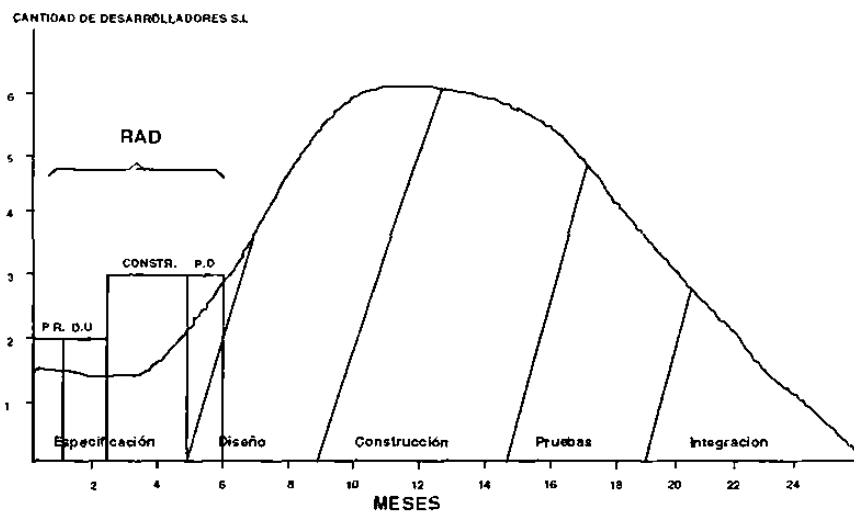


Fig. 3.7 Ciclo de vida clásico vs. RAD

Sólo que existe un inconveniente ya que para lograr esta reducción de tiempo se requiere primeramente una estructura dentro de la cual esté RAD. Esta estructura requiere de una arquitectura técnica y de información.

La primera permitirá que los usuarios tengan accesos a diferentes recursos de información ayudándolos a tomar mejores decisiones ya que a través de un estándar basado en sistemas abiertos permitirá que los componentes de diferentes proveedores sean conectados todos juntos sin ningún problema.

Y en cuanto a la arquitectura de información ésta será diseñada utilizando la ingeniería de información antes descrita, ya que así el desarrollo de sistemas con RAD será más rápido porque el modelo de datos ya existe.

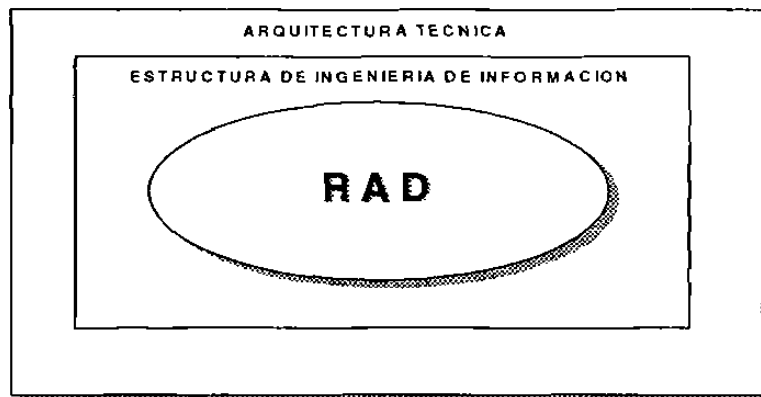


Fig. 3.8 Elementos indispensables para aplicar RAD

Una vez que esta estructura técnica y de información exista es importante contar con:

herramientas, gente, administración y metodologías.

Gente: Se debe contar con un equipo de personas especiales y altamente entrenadas, que conozcan cómo usar las herramientas y que trabajen juntos.

Dentro de la comunidad de sistemas de información al emplear RAD podrían mencionarse a:

- Administrador de proyecto.
- Equipo de construcción.
- El líder del taller de trabajo RAD.
- Escribiente.
- Experto en factores humanos.
- Experto en modelación de datos.
- Administrador del repositorio.
- Ejecutivo.
- Campeón del sistema.
- Coordinador de usuario.
- Equipo de planeación de requerimientos.
- Equipo de diseño de usuario.

- Equipo asistente de construcción.
- Administrador de entrenamiento.
- Consejo de usuarios de revisión.

Donde cada uno de ellos lleva a cabo su función con el fin de desarrollar un sistema que cumpla con los requerimientos del usuario en el momento de su puesta en operación.

Administración: Una buena administración adoptará nuevas herramientas y técnicas fomentando el entrenamiento y la capacitación entre su personal.

Para lograr ese cambio de cultura en las personas, el ejecutivo de Sistemas de Información deberá estar comprometido para lo cual, deberá entender la metodología para así poder implementarla.

Herramientas: RAD dependerá del uso de herramientas automatizadas. Las herramientas seleccionadas deberán soportar las técnicas en las que se apoya la metodología RAD, y si el sistema a construir es complejo requerirá de herramientas complejas.

Hoy en día es muy recomendable contar con una herramienta I-CASE que provea un conjunto integrado de herramientas soportando todo el desarrollo de un sistema, incrementando así la productividad y permitiendo la integración entre los diferentes componentes del proyecto. Esto se hará de tal forma que se cuente contando con una herramienta de este tipo; y así, diferentes equipos de personas podrán trabajar sobre un mismo proyecto por más complejo que sea este. Las personas deberán aprender a administrar el ciclo de vida RAD con herramientas de este tipo.

Metodología: Es indispensable contar con la metodología que especifique el proceso de desarrollo. Contar con una metodología permitirá que los desarrolladores conozcan la secuencia de tareas, a realizar, las entradas y salidas de cada tarea, el cómo realizar la tarea, la gente necesaria y el entrenamiento necesario para llevarlas a cabo, así como las pautas acerca de cómo tener éxito y que dificultades evitar.

La buena metodología será adaptable completamente a las circunstancias, minimizará el trabajo manual, asumirá que los desarrolladores son inteligentes y creativos, proveerá las pautas para el éxito, las advertencias de dificultades y una lista de control tal que las acciones importantes no se olviden.

Los resultados con RAD podrán observarse desde los primeros tres meses a diferencia con un desarrollo tradicional cuyos resultados son conocidos en por lo menos 18 meses, dependiendo del tipo del proyecto.

3.5. PROTOTIPEO.

3.5.1. DEFINICION.

El prototipeo es una técnica que nos permite construir rápidamente una versión tipo copia de un sistema deseado, o de partes de un sistema. Además de la técnica se incorpora el uso de poderosas herramientas de desarrollo de software para construir el prototipo identificando con rapidez, las necesidades particulares de información del usuario, logrando reducir el tiempo dedicado a la fase de análisis y diseño general.

3.5.2. TIPOS.

Los tipos de prototipos que actualmente se pueden generar son [JOH,91]:

- Bocetos: De salida de reportes impresos o pantallas en línea.
- Simulación: Interacciones simuladas de operaciones en línea y reportes batch. Este modelo de prototipo es utilizado en el ciclo de vida tradicional del sistema en la fase de diseño. Esta simulación involucra una relación limitada entre las transacciones.
- Modelo de trabajo: Sistema parcial con interacciones entre archivos y transacciones. Si se completa puede llegar a implementarse.
- Investigación y desarrollo. Esta reemplaza al ciclo de vida tradicional, quedándose sólo las fases de estudio de factibilidad, diseño detallado e implementación.

Los pasos que se llevan a cabo utilizando este tipo de prototipos son [POW,90]:

- Identifique los requerimientos de información básica del usuario.
- Desarrolle un modelo de trabajo.
- Utilice el modelo de trabajo para refinar necesidades y justificar el sistema.
- Perfeccione el modelo y posiblemente implementarlo.

3.5.3. USOS.

Es importante recalcar que se recomienda utilizar los prototipos para sistemas en línea o interactivos. Un prototipo resulta ser valioso si:

- Hay libertad para la creatividad del usuario para mejorar el sistema.
- Los usuarios muchas veces no están seguros de lo que realmente quieren.
- El sistema cambia una operación básica del negocio.
- Un diálogo de usuario-final debe ser probado con los usuarios para ver si puede ser mejorado.
- Los usuarios no entienden los impactos del nuevo sistema.
- Las funciones son sutiles y los usuarios las entienden mejor que los analistas.
- Las pantallas y los reportes deben ser checados con la administración para ver si ellos pueden hacerlos más útiles o fáciles de usar.
- Los usuarios tienen dificultad de expresar todos los requerimientos del sistema.
- El prototipo puede actuar como un catalizador para sacar ideas alternativas.
- La experimentación puede ser hecha para lograr mejores prácticas de negocio.

3.5.4. LAS HERRAMIENTAS PARA PROTOTIPOS.

Con lenguajes de cuarta generación y generadores de código, es posible construir un modelo de trabajo rápidamente. Una herramienta de prototipeo debe ser parte de un conjunto de herramientas I-CASE que permita diseñar, generar y optimizar el código final. Ya no se trata de realizar el prototipeo con un tipo de herramienta y el sistema con otra sino del prototipeo llegar a la construcción del sistema logrando cumplir los requerimientos de los usuarios tan rápidamente como sea posible.

Algunas de las características que se deben considerar si se va a utilizar una herramienta es que para prototipeo serían:

- Debe ser una parte integrada del conjunto de herramientas I-CASE.
- Utilizar diseño, templates y estructuras almacenadas en el repositorio.
- Almacenar los resultados del prototipeo en el repositorio I-CASE.
- Sea interactivo.
- Sea fácil de utilizar.
- El diseño de pantallas y reportes sea rápido.
- Realizar cambios fácilmente y rápidamente.
- Soportar estructuras de diálogo deseables, tales como barras de desplazamiento, operaciones con mouse, menús desenrollables etc. (Que permita crear una buena interfase gráfica)
- Soportar estructuras de B.D. apropiadas.
- Proveer fácilmente para pruebas.
- Dar un buen desempeño o funcionamiento de máquina.
- Que permita que el prototipo tenga una buena interfase de comunicación.

Ahora bien si su herramienta cuenta con las características antes mencionadas hay que determinar quién construirá ese prototipo. Es recomendable que en esa construcción sean dos personas las que intervengan: el usuario y un profesional de sistemas de información siendo éste último una persona hábil y rápida en el manejo de la herramienta. Una vez construido el prototipo este puede ser revisado por el mismo usuario hasta que el prototipo este completamente refinado por él , una última revisión se recomienda, en la cual podrán intervenir:

- Personal técnico encargado de construir el sistema.
- Administración.
- Algún consultor externo

El prototipo resulta al final de cuentas ser útil para que los usuarios se entrenen utilizándolo hasta que el sistema final este listo.

3.5.5. BENEFICIOS Y PELIGROS.

Beneficios.

- Los usuarios entienden y reaccionan a los prototipos mucho mejor que las especificaciones en papel.
- Con una buena herramienta se construye más rápido un prototipo.
- El prototipo introduce realidades. Los usuarios podrán observar y criticar lo que se está *construyendo*.
- Sin un prototipo existe un riesgo de construir un sistema inadecuado.
- El prototipo permite que los errores y debilidades sean atrapadas antes de un diseño caro y que una programación sea realizada.
- Los prototipos pueden generar emoción y mejorar la moral de los usuarios y de los desarrolladores.
- Son utilizados como herramientas de entrenamiento.
- Permiten comunicar los requerimientos a los programadores.
- Logran que se lleva a cabo un rápido desarrollo.

Peligros

- Rápido diseño informal puede reemplazar un diseño bien estructurado.
- El prototipo alienta a los usuarios a cambiar sus opiniones acerca de los requerimientos. Ellos pueden continuamente inventar los requerimientos tal que el prototipo constantemente cambie y no converga rápidamente a una forma implementable.
- Los usuarios pueden pensar que pueden tener el sistema inmediatamente. (Si sólo se han presentado los diseños de pantalla).
- Hay una tentación para construir el prototipo sin adecuada consideración de seguridad, auditabilidad, recuperación, reserva, mantenibilidad, desempeño, interconexión en redes o documentación.

3.6. JAD.

El JAD es una técnica desarrollada por la IBM en 1977 con el fin de ser utilizada en el diseño de sistemas para mejorar la calidad de las especificaciones de un futuro sistema, comprometiéndolo a los usuarios en el proceso de definición de requerimientos

Esta técnica logra grandes resultados como

- Mayor productividad

- Calidad en el diseño.
- Trabajo en equipo.
- Disminución de costos de desarrollo y mantenimiento.

Las fases que la componen son la adaptación al usuario, sesión y resumen [AUG,91].

Adaptación al usuario: Es la clave para un efectivo JAD. El líder de la sesión y uno o dos analistas JAD se preparan para las actividades de JAD/Plan y el JAD/Diseño. Las tareas que se llevan a cabo en esta fase como la orientación dirigida, organización de equipo, adaptación del JAD y preparación de materiales se logra definir quiénes serán los participantes para las otras fases así como la logística, los diferentes materiales a utilizar en la fase de sesión y la aclaración de dudas previas a la fase de sesión sobre algún tema en particular relacionado con el proyecto,

Sesión para el diseño del sistema: El determinar los objetivos del sistema y los requerimientos para crear un diseño externo constituye el fin de esta fase.

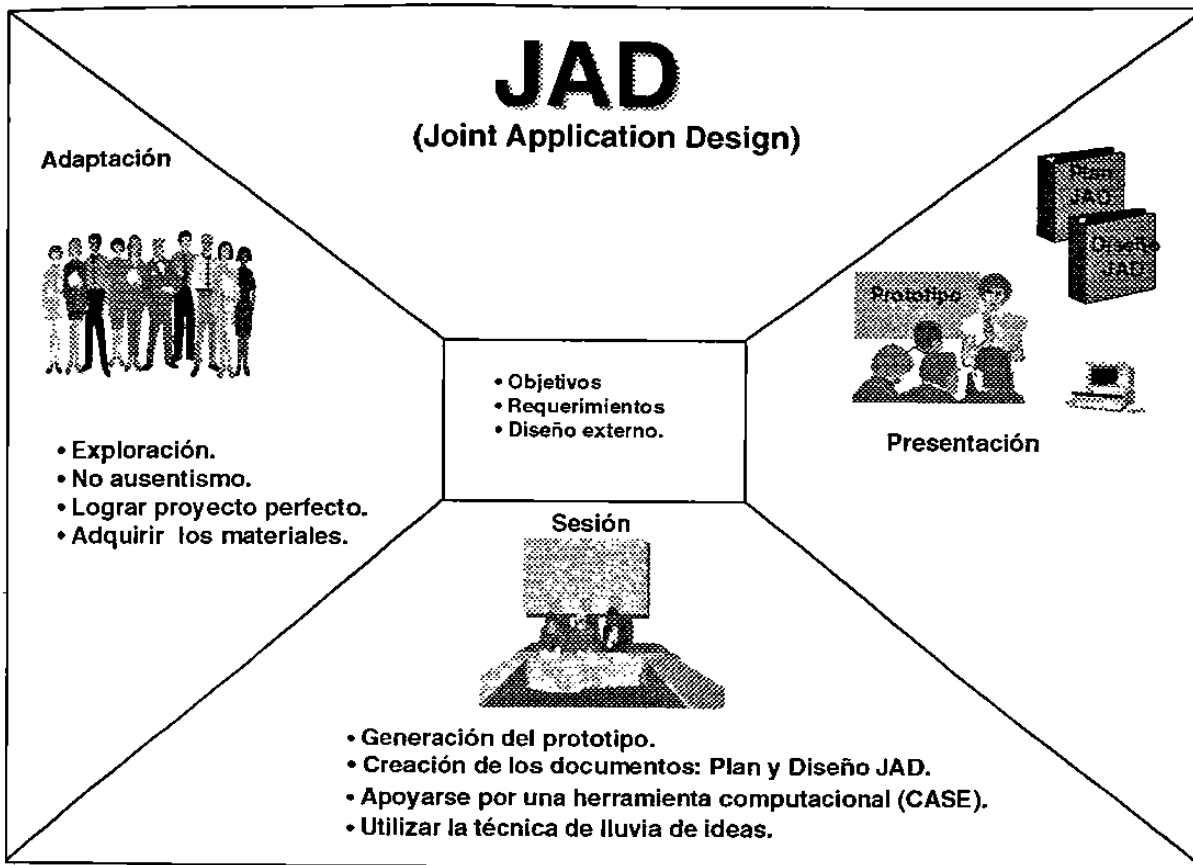
Las personas involucradas en esta fase deberán aprender el trabajo del usuario para posteriormente diseñar pantallas y reportes para el sistema. Esto permite que tanto la experiencia del usuario como de la persona de sistemas de información, trabajen en equipo construyendo el diseño.

Para poder lograr esa calidad buscada en el diseño se recomienda hacer a cabo las fases :

- Presentación de tareas.
- Generación de Ideas.
- Evaluación.
- Compromiso.

Resumen para el diseño del sistema: Un prototipo construido así como los resultados del diseño se presentan al ejecutivo, así mismo se generará el documento relacionado con el Plan de JAD y el Diseño de JAD los cuales reflejan las decisiones y los formatos que se generaron en la sesión, y serán distribuidos tanto a los participantes como al equipo técnico de personas para llevar a cabo el diseño interno del sistema, ya una vez actualizado el documento.

Los beneficios que representa seguir esta técnica los podemos mostrar con el siguiente esquema:



Involucramiento real del usuario

Eliminación de especificaciones en papel



Reducción de costos



Sistema con calidad



Fig 3.9 Fases de JAD y sus beneficios

4. CASE(Computer Aided Software Engineering).

Ya que CASE, es considerada como toda una filosofía para la modelación de negocios, sus actividades y el desarrollo de sistemas de información, esta filosofía ha involucrado el uso de la computadora como una herramienta para soportarlo. Toca el turno de enfocarnos específicamente a esas herramientas de CASE para lo cual conoceremos sus componentes, la manera como estos se han clasificado en el mercado, el proceso que se requiere para su implantación el avance tecnológico que se pretende con respecto a ellas y la realidad en algunas empresas más importantes de nuestra localidad con respecto a esta tecnología.

4.1. COMPONENTES.

Las herramientas CASE las cuales cubren las necesidades de los desarrolladores de sistemas deben contener diversos componentes que provean diferentes funciones de tal manera que logren soportar a todo el ciclo de vida de desarrollo de sistemas. Podríamos clasificar estos componentes de acuerdo a su función como es: la generación de documentación, realización de análisis, coordinación de proyectos y generación de aplicaciones.

Documentación.

- Generador de gráficas. Es utilizado para modelar representaciones del sistema como los diagramas de flujo de datos, diagramas estructurados, diagramas entidad relación, diagramas de flujo de programas. Permite además llevar a cabo ligas de los diagramas al diccionario de datos.
- Diccionario de datos. Repositorio central necesario para llevar a cabo la integración entre todos los módulos de la herramienta ya que contienen la información relacionada con bases de datos, archivos, estructuras y elementos, también como las especificaciones para los procesos, programas, reportes y pantallas.

Este diccionario puede ser accesado ya sea a través de su propia función o de cualquier otro componente CASE.

- Procesador de palabras. Algunas capacidades de edición de textos son incluidas en las herramientas CASE. Su función primaria es preparar la documentación como las especificaciones del diseño para los archivos, estructuras, registros y diagramas. Generalmente, son documentos o formas las que se proporcionan para llevar a cabo las especificaciones, y son simplemente templates de procesadores de palabras con formatos predeterminados a ser llenados con respecto a las diferentes entidades de diseño. Algunas herramientas CASE son capaces de importar u exportar a procesadores de palabras comerciales.
- Diseñadores de pantallas y reportes. A través de este componente los diseñadores pueden diseñar sus reportes y pantallas para que sean revisadas por los usuarios.

Realización de análisis.

- Proceso de verificación. Las herramientas CASE verifican las inconsistencias o cualquier error en los diferentes elementos de la documentación.
Estos procesos pueden estar ligados a funciones de depuración de compiladores de lenguajes de programación. Como la programación, algunas metodologías de análisis y diseño de sistemas pueden ser tan sistemáticas que la computadora puede en realidad determinar circunstancias dudosas, para lo cual CASE puede utilizar "inteligencia" para verificar la exactitud de el diseño.
- Optimización de datos. Técnicas de normalización que pueden ser ejecutadas sobre la colección de archivos y de datos del sistema.
- Prototipo. Se usa para realizar el modelo de sistemas de operación actual con el cual se ejecuten todas las funciones del sistema propuesto. La herramienta CASE es la que tiene la habilidad para organizar los diagramas de las pantallas del sistema en un orden que puedan originarse de un menú de selección.

Coordinación del proyecto.

- Medios de presentación necesarios para los miembros del equipo del proyecto con el fin de mantenerlos informados a ellos como a los usuarios técnicos y administradores acerca del progreso actual.
- Administración de proyectos. La herramienta CASE se encarga de implementar técnicas para planear, controlar y monitorear el desarrollo de sistemas. PERT y GANTT se encuentran disponibles en estas herramientas.
- Interfases de aplicación. Las herramientas CASE proveen una interfase para un medio ambiente específico de operación. Las especificaciones de registro y de campos pueden ser limitadas a formatos compatibles con lenguajes de aplicaciones específicos como COBOL, PL/I, C, DBASE o un 4GL.

Generadores de aplicaciones.

- Generadores de código. Algunos productos pueden producir pseudocódigo, código, módulos o programas directamente de modelos de diseño, prototipos y especificaciones.
Código que después de generarse deberá ser compilado.
- Generadores de sistemas. Algunas herramientas CASE son capaces de ser el medio ambiente de aplicaciones, esto es de proveer la coordinación y las herramientas necesarias para todas las actividades del desarrollo de sistemas incluyendo la instalación y el mantenimiento. De tal manera que el modelo creado por la herramienta CASE llegará a ser el sistema.
- Interfases 4GL. Algunas herramientas CASE proporcionan una interfase a un lenguaje de cuarta generación, tal que un lenguaje de este tipo puede ser integrado en combinación con el diccionario de datos y un sistema administrador de base de datos.

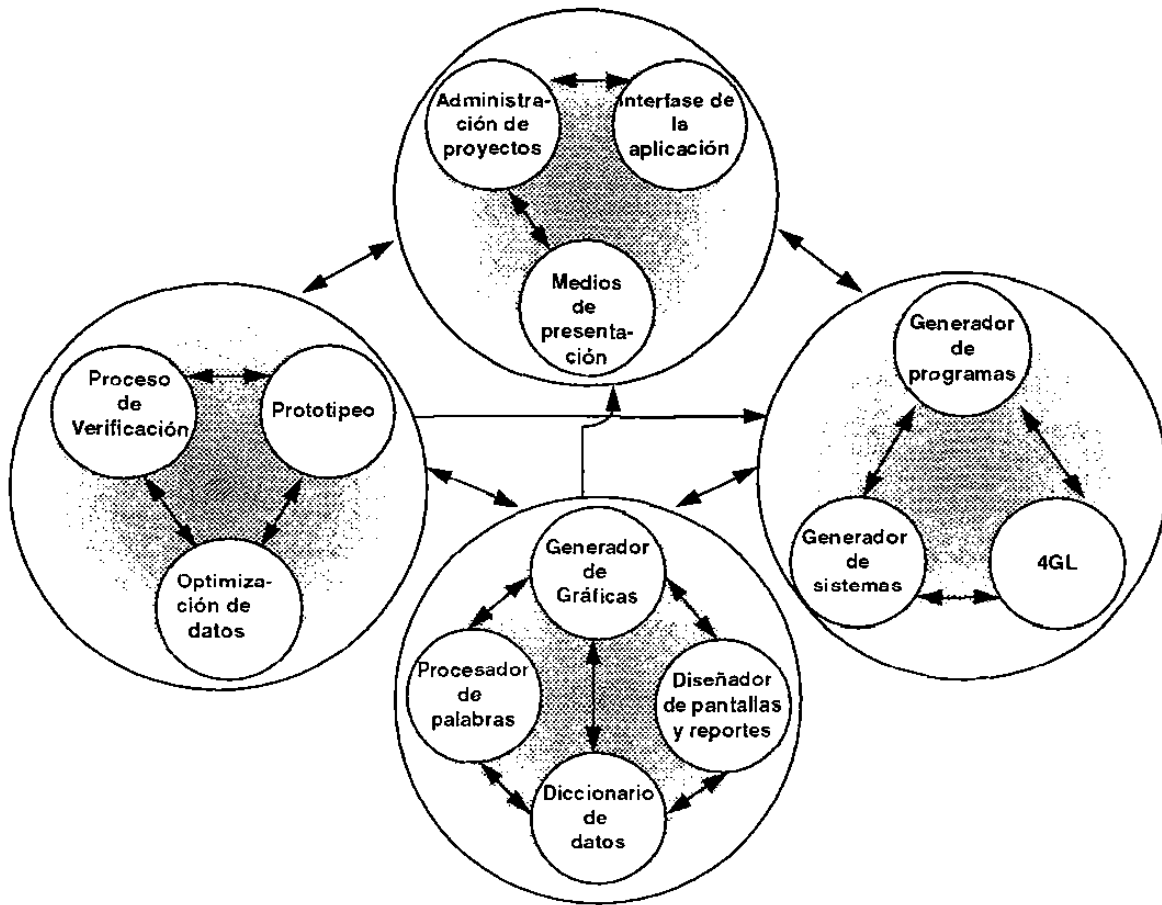


Fig. 4.1. Diagrama que muestra todos los componentes integrados

Por lo tanto la herramienta CASE perfecta lo será si proporciona un banco de trabajo (workbench) para coordinar todos los aspectos del desarrollo de sistemas, desde la planeación hasta su instalación y en el caso de sistemas ya desarrollados la re-ingeniería.

4.2. CLASIFICACION.

La clasificación que se le ha dado en su concepto como herramienta ha sido muy variada como lo son [PRE,93]:

- Por su costo.
- Por la arquitectura que la soportan.
- Por su función.
- O de acuerdo a la cobertura de las fases del ciclo de vida de desarrollo de sistemas

Explicando más a detalle estas dos últimas, vemos que en el caso de la función que éstas desempeñan tenemos:

- Herramientas de planificación de sistemas administrativos.
En base al modelo de requisitos de información estratégica se genera un metamodelo obteniendo sistemas de información específicos para la organización.
- Herramientas de administración de proyectos.
Permiten estimar costos, duraciones, definir una estructura y dividir el trabajo realizando un plan y dando seguimiento en forma continua.
- Herramientas de soporte.
Herramientas para documentación, administración de redes, administración de B.D., control de calidad y de configuración de software.
- Herramientas para análisis y diseño.
Permiten realizar la modelación de datos, de procesos, definir el diccionario de datos y realizar las representaciones de los procesos.
- Herramientas de integración y pruebas.
 - + Adquisición de datos.
 - + Medida estática.
 - + Medida dinámica.
 - + Simulación.
 - + Administración de las pruebas.
 - + Funcionalidad cruzada.
- Herramientas de creación de prototipos.
Hay una serie de herramientas que son consideradas de este tipo como serían aquellas que crean prototipo en papel cuya herramienta de dibujo se encuentra implementada en una PC o una estación de trabajo; o las herramientas de diseño de pantallas que ayudan a presentar los formatos de pantallas y sin pasar por alto aquellas que permiten crear el formato de los reportes.
- Herramientas de mantenimiento.
Este tipo de herramientas se encuentra clasificada en :
 - + Herramientas de ingeniería en reversa a especificaciones las cuales a partir del código fuente generan módulos de análisis y diseño estructurado.
 - + Herramientas de reestructuración y análisis de código: A partir de la sintaxis de un programa la cual se analiza generar un programa estructurado.
 - + Herramientas de re-ingeniería del sistema: Modifican sistemas de bases de datos.

Para continuar con la clasificación que a éstas se les ha dado, toca el turno a : según las fases del ciclo de vida de desarrollo de sistemas dividiéndolas en:

- Toolkits. Consideradas como el conjunto de herramientas integradas diseñadas para trabajar conjuntamente y automatizar total o parcialmente, el ciclo de vida de desarrollo de software.

- Workbenches. Herramientas integradas diseñadas para trabajar en conjunto y automatizar totalmente el ciclo de vida de desarrollo de software.
- Frameworks. Herramientas que permiten ligar las herramientas CASE con herramientas que no son CASE.
- Compañeros de metodologías. Conjunto de herramientas que automatizan las tareas en una metodología CASE particular.

Por último es importante incluir la clasificación que autores como McClure, Powers, Everest, Patrinostro han dado a este tipo de herramientas que pueden ser o no integradas[MCC,93], [POW,90], [PAT,92].

La primera clasificación que dieron fue:

- Herramientas Front-end o Upper CASE.
Cuyas funciones son soportar la planeación, análisis y diseño lógico; soportar las técnicas de diagramación y proporcionar conexión al repositorio.
- Herramientas Backend o Lower CASE.
Para soportar el diseño físico, proveer la generación de esquemas de bases de datos, llevar a cabo la generación lógica de procesos, controles y pantallas y lograr la conectividad al repositorio.

Después se realizó una división a las herramientas Upper CASE quedando la siguiente clasificación :

- Herramientas Upper CASE cuya función es ayudar al desarrollo y mantenimiento de los planes de sistemas de información. Asimismo ayudan a identificar las entidades de la organización, realizando referencias cruzadas entre las entidades para identificar interacciones; permitiendo también identificar funciones, actividades, metas y objetivos del negocio, así como usuarios problemas, oportunidades y decisiones críticas.
- Herramientas Middle CASE permiten identificar todos los procesos dentro del sistema, los datos que se utilizarán y su flujo a través de los procesos identificando además todos aquellos eventos que intervienen en la realización del proceso.
Dan soporte además a la modelación de procesos y datos.
- Herramientas Lower CASE son los generadores de aplicaciones y de prototipos así como aquellas herramientas que ayudan con la fase de construcción.
Realizan el mantenimiento a sistemas desarrollados o no con CASE utilizando las herramientas de re-ingeniería.

Por último es importante considerar aquellas herramientas conocidas por I-CASE cuyo concepto encierra a una única herramienta que automatiza todo el ciclo de vida de desarrollo de un sistema o representa un conjunto de herramientas, donde cada una de ellas soporta a una fase del ciclo de desarrollo comunicándose entre sí y soportando así, todo el desarrollo del sistema. Dentro de sus características principales se encuentran

- Apoyar a todo el ciclo de vida de desarrollo de sistemas.

6278,

- Manejar un repositorio que almacene el conocimiento de todas las herramientas de una forma integrada.
- Almacenar el conocimiento en el repositorio el cual checa automática y meticulosamente *asegurando la integridad y consistencia*.
- Generación automática de la documentación técnica.
- Manejo de una interfase de usuario consistente y uniforme para lo que son todos los módulos de la herramienta.
- Compartir información, permitiendo así la comunicación entre distintos desarrolladores.

Sin embargo son pocas herramientas con este enfoque de I-CASE ya que la mayoría no permiten que los resultados de un componente sean transferidos al siguiente (integración vertical), resolviendo así únicamente los síntomas de algunos problemas específicos.

4.3. HERRAMIENTAS ACTUALES EN EL MERCADO INTERNACIONAL.

Existen muchas herramientas CASE que pueden soportar una o más fases del ciclo de vida del desarrollo de un sistema, son sólo pocas las que soportan todo el ciclo las cuales pueden correr en diferentes plataformas. He aquí algunas de ellas. Si se requiere una en especial para una determinada etapa del CVDS ver anexo A.

CASE INTEGRADO

Compañía	Producto	S.O.	Hardware		
			Micro	Mini	Mainframe
Andersen Consulting	Foundation	DOS, OS/2, PM VAX/VMS, MVS CGOS.	■	■	
CGI Systems Inc.	PacBase	Windows, DOS, MVS, Unisys, UNIX, Tandem.	■	■	
	PacLAN	Windows, DOS, OS/2.	■		
	PacLAN/X	UNIX.	■	■	
Oracle Corp.	Oracle Case Tools	DOS, OS/2, PM, VAX/VMS, UNIX, MVS, Wkst.	■	■	
Computer Systems Advisers Inc.	X/CASE	DOS, UNIX, Wkst, NCR.	■	■	
	Adelia			■	
Cortex Corp.	CorVision	VAX/VMS, DOS	■		
SPS Software Products & Services Inc.	EPOS	DOS, VAX/VMS, UNIX, Wkst, DG Concurrent	■	■	■
Teledyne Brown Engineering	TAGS	VAX/VMS, UNIX, Wkst	■	■	
Texas Instruments Inc.	Information Engineering Facility	DOS, OS/2, VAX/VMS, UNIX, MVS	■	■	

Tabla 4.1 Herramientas que soportan todo el ciclo de vida de desarrollo de un sistema.

4.4. VENTAJAS Y DESVENTAJAS.

Algunas de las ventajas que ofrece CASE a las personas que se dedican al desarrollo y/o mantenimiento de sistemas son [MCC,93], [EDW,93]:

- Permite que se practiquen las técnicas estructuradas.
- Impone el uso de la ingeniería de software y de información.
- Se mejora la calidad del software mediante la comprobación automática, logrando que más usuarios estén satisfechos.
- Permite la realización de prototipos.
- Reducción del mantenimiento de programas.
- Acelera el proceso de desarrollo.
- Libera al desarrollador de la principal parte creativa en el desarrollo de software.
- Ayuda a que el desarrollo sea evolutivo e incremental.
- Permite que se reutilicen los componentes del software.
- Incrementa la productividad de los analistas y programadores.
- Mejora la comunicación entre los usuarios y los analistas.
- Logra una documentación técnica y del usuario consistente y completa.
- Hace cumplir estándares.
- Se decreta el tiempo de pruebas.
- Con CASE el porcentaje de desarrollo se dedica más al análisis y diseño y no tanto a la construcción y mantenimiento.
- La interfase y documentación es común para todos los sistemas.

Desventajas.

- No existe ninguna herramienta que proporcione todas las ventajas que se mencionaron.
- Alto costo.
- Consumen muchos recursos de hardware.
- Se requiere de tiempo para la formación y capacitación del personal.
- Personas dedicadas al desarrollo de software con metodologías de desarrollo personal se niegan a la innovación tecnológica. (Resistencia al cambio).

4.5. LA IMPLEMENTACION DEL CASE.

A pesar de las ventajas que proporciona la tecnología CASE, la experiencia ha demostrado que los resultados obtenidos por CASE no han sido en muchos casos los deseados, ya que antes de comprar CASE hay que pensar si realmente se necesita.

Muchas han sido las causas que han ocasionado el fracaso de CASE en la empresa como:

- Confusión de lo que realmente hace cada producto CASE individualmente.
- Utilizar herramientas CASE en problemas para los que no están diseñados.
- Poner demasiado empeño en las herramientas CASE como la solución completa.
- Ignorar la importancia de la buena administración.
- No tener metodologías o estándares de desarrollo.
- Herramientas CASE pobremente integradas.
- Herramientas pobres en documentación y capacitación.
- Falta de funcionalidad suficiente en las herramientas CASE.
- Falta de claridad sobre cuáles problemas de software necesitan solucionarse.
- Falta de uso de métodos para medir el impacto de CASE en el desarrollo y mantenimiento.
- Falta de capacitación en metodologías de desarrollo de software.
- Dificultad y vacilación con respecto a tomar una decisión sobre cómo utilizar la tecnología CASE.
- No se desea cambiar la manera actual de desarrollo y mantenimiento de software.
- Se ve a CASE como una tecnología de alto riesgo.
- Carencia de un plan para implementar la tecnología CASE.

El hecho es que para implementar la tecnología CASE con éxito en el empresa ésta debe estar preparada tomando antes en consideración :

- Que CASE no es una panacea.
- CASE deberá estar incluido en un plan estratégico de informática.
- CASE requiere el apoyo de la alta administración.
- CASE es compromiso de parte de todos los involucrados con el desarrollo del proyecto.
- CASE requiere entrenamiento.
- CASE implicará gastos más de una sola vez.

- CASE representará un total cambio cultural lo cual implicará que en vez de permitir que los desarrolladores de sistemas expresen sus impulsos creativos sin importar la productividad y la calidad del sistema, estos tendrán que convertirse en analíticos de negocios descubriendo así las reglas del negocio, para lo cual harán uso de diversas metodologías y técnicas antes olvidadas.

Si la empresa ha decidido implementar una tecnología CASE después de haber previsto los factores anteriores y está consciente de que cubrirá las necesidades de la organización es indispensable contar con un modelo o enfoque para implantar CASE.

A continuación se presentarán varios enfoques y modelos para implementar CASE.

El enfoque de Lucas Gibson.

El nos indica que para poder implantar CASE con éxito debemos seguir la siguiente pauta [GIB,89]:

- a) Establecer y aplicar un conjunto de criterios para evaluar y determinar cuál herramienta CASE adquirir.
- b) Incluir un entrenamiento extensivo proporcionado por grupos de consultoría CASE como parte de la planeación estratégica para implementar sistemas CASE.
- c) Seleccionar un grupo experimentado para proporcionar consultoría, entrenamiento y dirección durante la implementación.
- d) Formular un conjunto de estándares y procedimientos para el uso de CASE dentro del medio ambiente de computación de la empresa.
- e) Seleccionar un grupo de consultoría para llevar el entrenamiento de CASE para el medio ambiente, estándares y procedimientos actuales.
- f) Seleccionar un grupo experimentado de analistas de software, siendo el primer grupo para recibir entrenamiento.
- g) Si la meta es tener entrenamiento CASE y hacerlo internamente, el personal de entrenamiento existente debe estar incluido en el grupo de aprendices.
- h) Antes de probar las herramientas CASE con un proyecto real conducir al grupo a un completo conjunto de cursos de entrenamiento que los familiarice a ellos con los mecanismos de las herramientas CASE, y los enseñe cómo aplicarlos dentro del medio ambiente de computación de la empresa.
- i) Para aplicar CASE en un proyecto real, seleccionar una aplicación simple (proyecto) que pueda ser terminada de 3 a 6 meses.
- j) El grupo deberá ser entrenado por la gente de consultoría todo el tiempo que el proyecto sea soportado por CASE.
- k) Reevaluar y modificar los estándares y procedimientos de la empresa acorde a la experiencia en ingeniería de software.

- l) Antes de la siguiente ola de entrenamiento, es importante refinar los cursos de entrenamiento para ajustarse más al nuevo medio ambiente CASE en la compañía.
- m) Entrenar a todo el staff debería de ocurrir en un período de 6 meses a 2 años, dependiendo del tamaño del staff.
- n) Ya teniendo la segunda ola de aprendices se comienza a entrenar.
- ñ) Los diferentes niveles de experiencia se involucran en ser entrenados y en conducir entrenamientos, el grupo de consultoría deberá ayudar al personal de casa a entrenar a la segunda ola de aprendices.
- p) Por último para la segunda ola de aprendices, el grupo de consultoría debe entrenar a cada equipo de proyecto, guiado por un miembro del grupo, todo el tiempo en el primer proyecto de sistemas.

Modelo de etapas de Huges y Clark.

Ellos nos dicen que para lograr la implementación exitosa de CASE en una organización ésta debe de pasar a través de varias etapas, las cuales se dan a conocer en su modelo [HUG,90].

Las etapas de este modelo son: desencantamiento, resignación o renuncia, compromiso, implementación y madurez. Representando así el camino que una organización tomará para llegar a implementar CASE en su medio ambiente de sistemas de información.

Cada una de las etapas es acompañada de un punto iniciador o evento que señala el movimiento de una etapa a la siguiente.

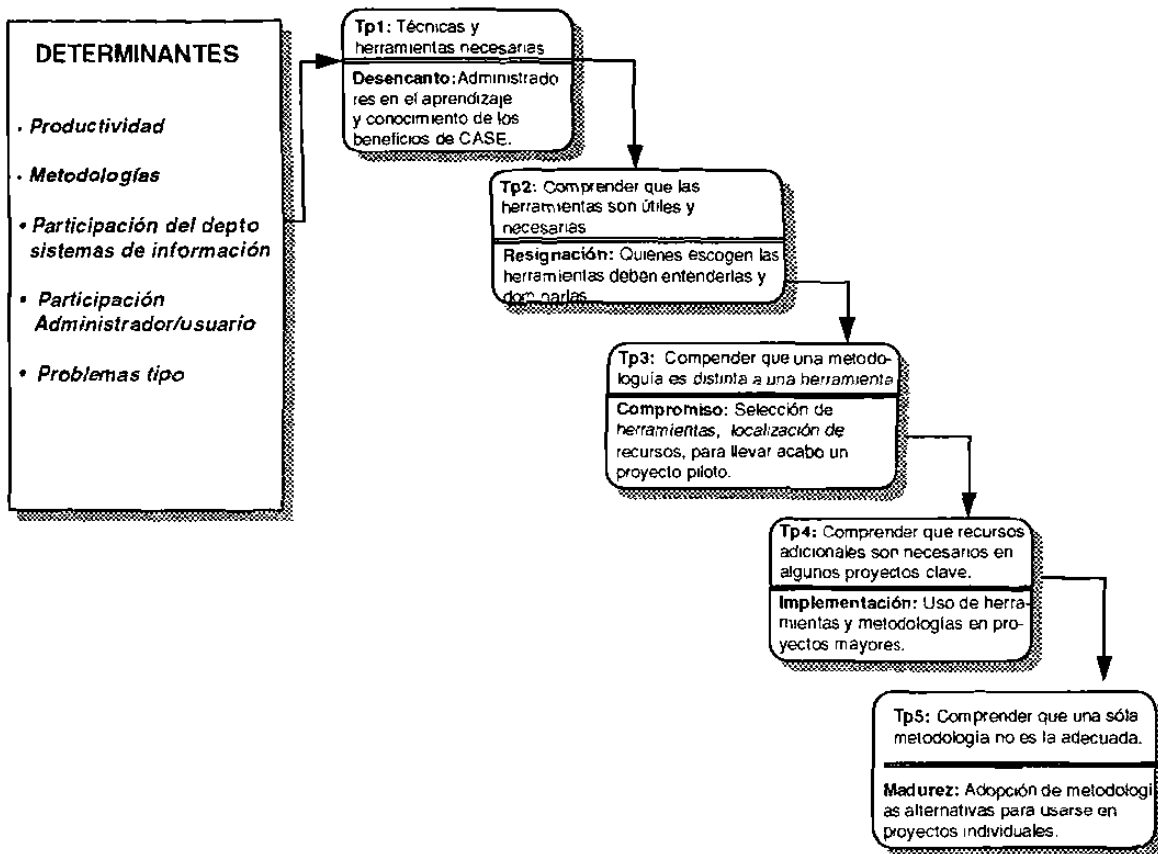


Fig. 4.2 El modelo de etapas para implantar CASE según Huges y Clark

Como se aprecia en la figura anterior existen determinantes como:

- Productividad.
- Metodologías.
- Participación del departamento de sistemas de información.
- Participación Administrador/ usuario.
- Y los problemas tipo.

En el caso de la productividad, ésta contiene dos criterios para evaluar el comportamiento organizacional: *eficiencia y eficacia*.

Al principio en el modelo de etapas, el énfasis es hacia la eficiencia. Esto puede resultar por la selección de herramientas CASE que comprimen el ciclo de vida, incluyendo lenguajes de cuarta y quinta generación, así como los generadores de código; poniendo posteriormente más énfasis a la eficacia.

Metodologías.

Las metodologías cuales han tenido limitada funcionalidad, al sólo soportar las primeras etapas de desarrollo de sistemas. Gracias a la funcionalidad completa de CASE permiten una firme búsqueda de la etapa de madurez en el modelo. Los elementos de la funcionalidad abarcan: planeación estratégica, modelación de datos, diseño de sistemas, construcción, administración de sistemas, soporte a proyectos, administración de la configuración, ingeniería en reversa, aseguramiento de la calidad, métricas y mantenimiento.

- Funciones de la Metodología
- Planeación Estratégica
 - Análisis
 - Modelación de datos
 - Diseño del sistema
 - Construcción
 - Administración de Sist.
 - Soporte a Proyectos
 - Administración de Conf.
 - Ingeniería en Reversa
 - Aseguramiento Calidad
 - Métricas
 - Ma.

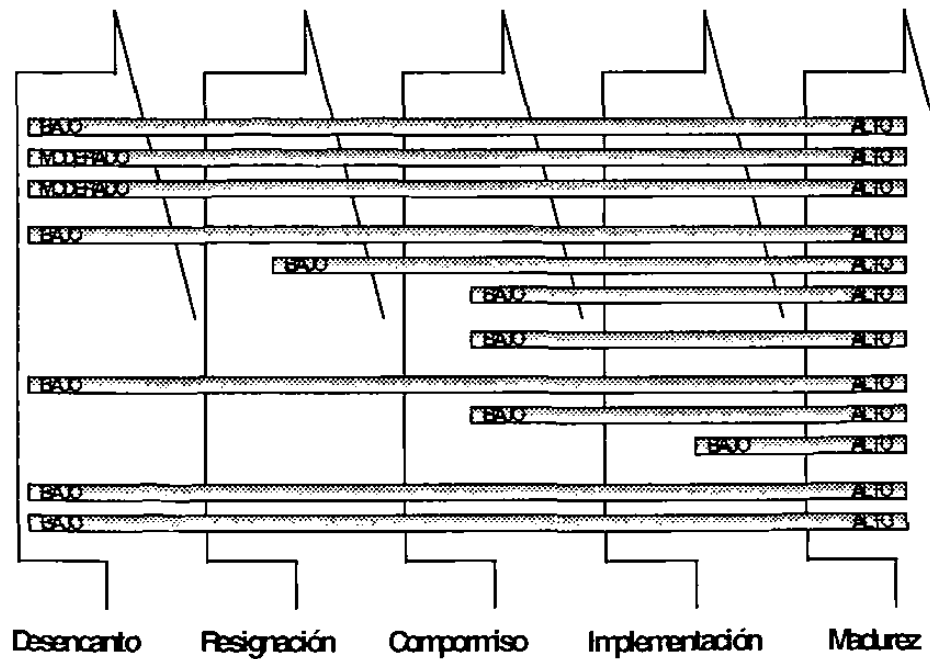


Fig.4.3 El modelo de etapas y su relación con las funciones de una metodología.

Participación de sistemas de información.

La participación de sistemas de información es una categoría de determinantes que intentan capturar la madurez del staff de desarrollo de sistemas.

A continuación se apreciará el grado de involucración del personal de sistemas de información en las diferentes etapas.

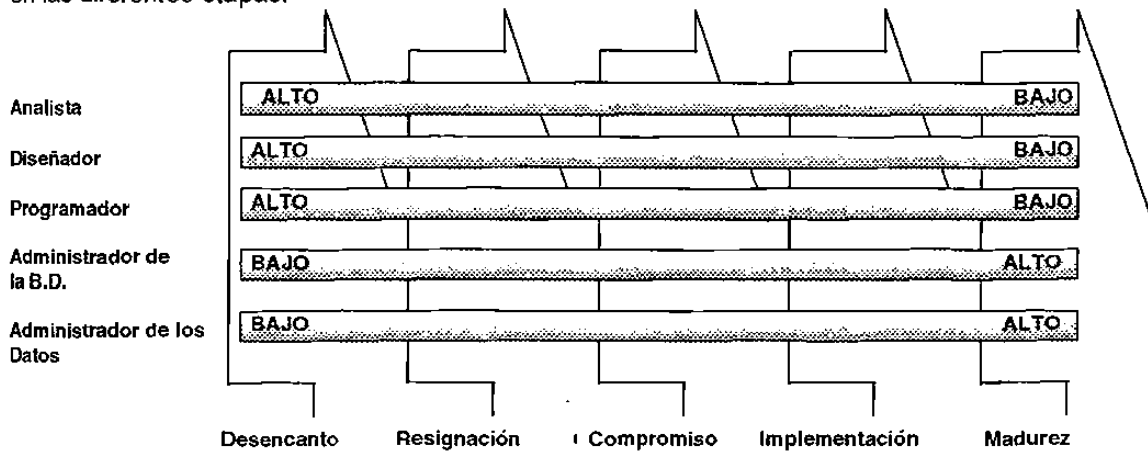


Fig. 4.4 Involucramiento del personal de sistemas en las etapas de implementación de CASE

Involucramiento ; Administrador/usuario.

Esta es una categoría de determinantes que se enfoca en los niveles organizacionales, la participación del usuario e influencia política, como lo son:

Las funciones de control operacional: que dominan el desarrollo de sistemas en las primeras etapas del modelo y han incrementado su importancia en términos absolutos a través de las etapas.

Control Administrativo: filosofía soportada en la organización durante las primeras etapas , ésta no afecta grandemente sino hasta la etapa de resignación. Desde aquí la mayor importancia es lograda.

Planeación estratégica: entra al modelo de etapas durante la etapa de implementación y llega a ser muy importante en la madurez. En realidad una de las justificaciones principales de CASE es que logrará que se tenga una ventaja estratégica.

El involucramiento de los usuarios: Este empieza en el desarrollo de sistemas con un nivel bajo de actividad durante la etapa de desencanto y se incrementa durante la madurez. La ventaja de la tecnología CASE es que provee una útil interfase con los usuarios.

Políticas: Siguen el mismo patrón que el involucramiento de los usuarios.

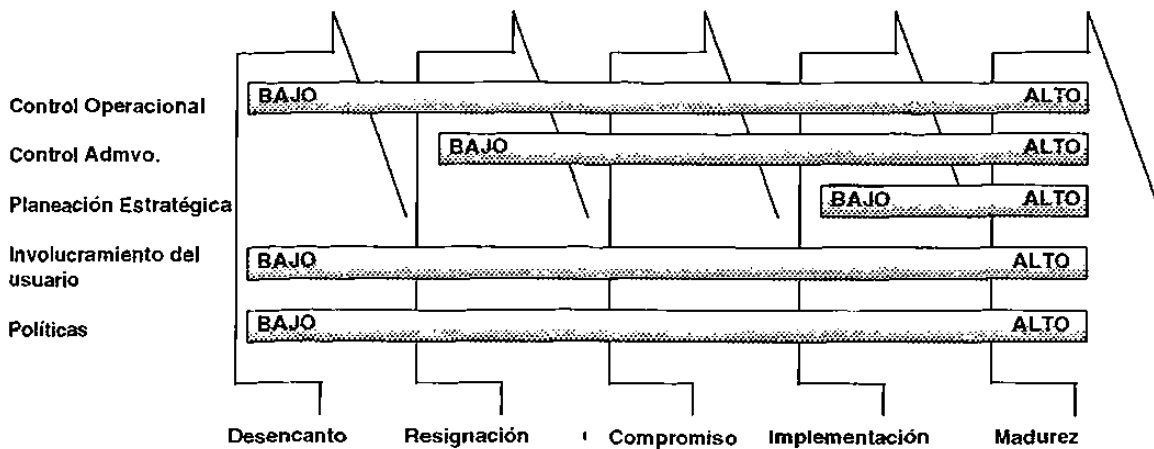


Fig. 4.5 La organización en las etapas de implementación de CASE

Problemas tipo.

Las primeras etapas del ciclo de vida son dominadas por soluciones algorítmicas. Como uno progresa a través de las siguientes etapas se pueden emplear más heurísticas. Las empresas en las etapas posteriores entienden la utilidad de tanto algoritmo y soluciones heurísticas y se concentran

aplicándolas en forma simple o compuesta como la garantía condicional.

La realidad es que con este modelo las firmas pueden determinar en cual etapa de CASE ellos están. De aquí que ellos puedan tomar decisiones que puedan acortar o eliminar etapas.

Con el modelo la organización evitará dificultades en la implementación del CASE. Esto ayudará a poner y hacer cumplir políticas, procedimientos y estándares.

Finalmente, lo más importante será un entendimiento del modelo de etapas permitiendo a la empresa más rápida y exitosamente implementar cualquier tecnología no sólo CASE.

Modelo de Fisher.

Para Fisher la clave en cualquier organización para poder asimilar una nueva tecnología es la presencia de un campeón de la tecnología. Ya que debido a la inercia que sufren las organizaciones realmente necesitan de una fuerza para mandar a otra dirección su enfoque.

El recomienda los siguientes pasos para la asimilación de la tecnología CASE [FIS.88]:

- a) Nacimiento de un campeón.
- b) Educación y entrenamiento.
- c) Evaluar la herramienta en la empresa.
- d) Selección del primer proyecto.
- e) Uso de la herramienta en el primer proyecto.
- f) Evaluación del primer proyecto.
- g) Preparación para proyectos grandes.



Fig. 4.6 Proceso para asimilar CASE

- Para que nazca, un campeón primero es necesario reunir información acerca de los proyectos exitosos que usaron CASE en otras compañías u otros desarrollos dentro de la propia compañía.

La otra opción será persuadir a la administración para invertir, tiempo, dinero y esfuerzo para investigar la tecnología.

Todo esto es con el fin de conocer algunas personas en la organización que estén decidiendo apoyar la nueva tecnología.

- Educación y entrenamiento. Este proceso ocurre gradualmente sobre el tiempo. Primero se contacta algunos vendedores y se les solicita literatura sobre sus productos. El segundo paso es asistir a conferencias de ingeniería de software donde la tecnología CASE se discute. Finalmente tomar un curso de entrenamiento ofrecido por un proveedor o una compañía comercial de entrenamiento, para lo cual se sugiere tomar varios cursos, solicitar una o dos herramientas para evaluarlas y tomar en cuenta la opinión de un grupo de consultoría externa con suficiente experiencia. El campeón puede realizar presentaciones a grupos interesados en el desarrollo de software y a la administración, e invitar a proveedores a realizar presentaciones de sus herramientas y tecnología en general con el fin de convencer a otros de la oportunidad.
- Evaluación interna de la herramienta. Este es el primer paso en el proceso de asimilación el cual representa un riesgo real tanto para el campeón como para la organización.

El campeón seleccionará varias herramientas para evaluar, con el fin de que se eduque a la organización y además se lleve a cabo la mejor decisión de compra. Se requiere por lo menos de un mes de evaluación para cada herramienta, con el propósito de que se realice una evaluación detallada para tomar una mejor decisión de compra, así como Tome en cuenta el entrenamiento y el soporte que da el proveedor y evalúe conforme alguna guía de selección de herramientas CASE.

- Selección del primer proyecto. Seleccionar el primer proyecto tiene como objetivo hacer que el proyecto tenga éxito en menos tiempo y con menos costo que sin CASE. Las siguientes cuestiones deben considerarse cuando exploramos por un buen primer proyecto:
 - + Seleccionar un nuevo proyecto para iniciar (No sobre un proyecto ya en desarrollo).
 - + El líder del equipo del proyecto debe ser un campeón CASE.
 - + La administración debe participar.
 - + Seleccionar un proyecto de tamaño pequeño a mediano.
 - + Seleccionar un proyecto normal.(No riesgoso).
 - + Seleccionar un proyecto significativo y redituable.
 - + Asegurar la cooperación de los usuarios.
 - + Capacitar al equipo del proyecto en las herramientas y metodologías apropiadas.
- Utilizar una herramienta en el primer proyecto implicará:
 - + Enfocarse en los requerimientos y en los detalles del diseño.
 - + Más tiempo en las entrevistas con los usuarios.
 - + El periodo de diseño se alargará.

- + Varias iteraciones durante el proceso de diseño.

Será necesario cuantificar y medir para evaluar, tomando en cuenta las siguientes métricas:

- + Duración de los períodos de implementación e integración de sistemas.
- + Total de personas por mes para los esfuerzos de desarrollo que se requieren para una versión aceptable del sistema.
- + Número de requerimientos de cambio del diseño por parte de los usuarios durante las pruebas beta.
- + Número de requerimientos de cambios del diseño por parte de los usuarios después de las pruebas.
- + Número de errores por línea de código.

Exitosamente se ha completado el primer proyecto de desarrollo basado en CASE. Siguiendo después con un entrenamiento al resto de la organización en metodologías y herramientas CASE. Se debe de considerar que la capacitación sea continua para todos los nuevos miembros.

Y ésta será la forma de institucionalizar la tecnología CASE.

McClure y su enfoque.

Si se va a adoptar la tecnología CASE debe haber un plan de implantación que se lleve cabo por fases el cual incluye [MCC,89]:

- Consideraciones administrativas.

- + Aceptar la adquisición de la herramienta CASE, la capacitación y el apoyo continuo.
- + Establecer objetivos y presupuestos.
- + Aceptación de la tecnología CASE por la dirección y los profesionales de desarrollo y mantenimiento.
- + Crear un grupo para implantar y dar soporte a la tecnología CASE.
- + Asignar responsabilidades para la investigación, implantación y evaluación de CASE.
- + Obligatoriedad de estándares y procedimientos CASE.
- + Crear programas de formación sobre herramientas CASE y otras metodologías.
- + Definir tiempos para cada paso en la implantación de CASE.

- Consideraciones técnicas.

- + Selección de la metodología de desarrollo.
- + Definición de los estándares técnicos para interfases con herramientas, sistemas administradores de bases de datos, lenguajes de cuarta generación, diccionarios.
- + Definición de las capacidades de la herramienta CASE.
- + Definición de las necesidades del hardware y de las redes de área local.
- + Selección de las herramientas CASE para que satisfagan las metodologías estándares y procedimientos de la organización.
- + Definición de la formación técnica necesaria.
- + Definición de las necesidades de soporte técnico para las herramientas y metodologías CASE.
- + Definición de los tipos de sistemas a desarrollar y mantener con las herramientas CASE.

- + Realización del proyecto piloto con CASE.
- + Gestión de la configuración de la herramienta CASE.

Para probar la herramienta CASE elegida se deberá primero elegir un proyecto piloto real y no crítico, definiendo los objetivos de cómo utilizar y medir el éxito de CASE antes de la prueba.

Los participantes en este proyecto piloto deben ser profesionales de desarrollo deseosos de utilizar la tecnología CASE.

Al terminar los resultados obtenidos éstos pueden utilizarse para:

- Sugerir que herramientas y metodologías son más útiles.
- Convencer a la dirección o administración y a los desarrolladores de los beneficios que se logran al usar CASE.
- Definir estándares y procedimientos que pueden utilizarse para aplicar la tecnología CASE.

La implantación de CASE suele asignarse a un grupo de profesionales del software cuyas responsabilidades consisten en:

- Investigar nuevas tecnologías.
- Elaborar un plan para implantarlas.
- Proporcionar un apoyo continuo para la nueva tecnología.

El éxito realmente de CASE se logrará si se consigue vender la idea a los directivos y profesionales de software en la empresa. Si no llegara a comprenderse la necesidad de un cambio y la mejora en el proceso de software con CASE, no se aceptará.

McClure señala diferentes formas de vender la tecnología CASE:

- Con proyectos pilotos que demuestren que CASE funciona en la organización.
- Con medidas que demuestran el impacto positivo de CASE.
- Con la participación de los usuarios de CASE en el proceso de selección de la herramienta y en la formulación de los estándares y procedimientos.
- Con la formación en las herramientas y metodologías estructuradas CASE.
- No deteniendo nunca la venta.

4.6. CARACTERÍSTICAS DESEABLES PARA UNA METODOLOGIA DE DESARROLLO DE SISTEMAS.

Si se está pensando en obtener una nueva metodología de desarrollo de sistemas la cual quizás esté acompañada de una herramienta CASE de sistemas, deberán considerarse las siguientes características :

- La metodología debe cubrir todo el ciclo de desarrollo de sistemas.
- Debe facilitar transiciones entre las fases del ciclo de desarrollo.
- Debe también soportar correcciones al sistema en todo el ciclo de desarrollo.
- La metodología debe soportar a la organización del desarrollo de software, para dirigir a los desarrolladores, así como permitir que trabajen juntos.
- Deberá ser utilizada por una amplia clase de proyectos de software.
- Deberá estar soportada por herramientas automatizadas que mejoren la productividad de un desarrollador o de un equipo de desarrollo.
- Por los cambios que puedan ocurrir en el sistema debiera soportarlos.

4.7 ELEMENTOS INDISPENSABLES PARA LA SELECCION DE CASE.

Si se ha decidido utilizar una herramienta CASE deberá contar con una metodología de desarrollo y una arquitectura para la integración de la herramienta antes de su instalación.

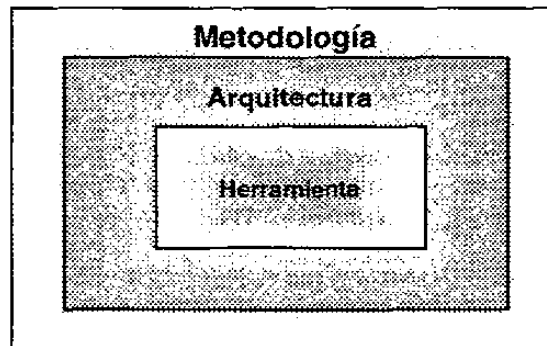


Fig. 4.7 Elementos indispensables para la selección de CASE

Debido a la gran variedad de herramientas en el mercado que son CASE, o que dicen ser CASE, necesitamos que los desarrolladores de sistemas analicen sus áreas de importancia y en base a sus necesidades elijan la herramienta que más les convenga. Para decidir sobre qué herramienta CASE es la que más conviene a la empresa, podemos considerar los puntos de vista que hacen diferentes autores.

Los factores de evaluación para una herramienta CASE recientemente publicados por *Computerworld*, estarán acompañados de una serie de interrogantes que debemos plantearnos si queremos determinar si la herramienta que estamos evaluando cumple con esa característica:

- a) Facilidad de uso.
Claridad de pantallas y comandos.

¿Es posible someter a proceso batch trabajos tales como las compilaciones mientras los desarrolladores trabajan interactivamente? ¿Afectan las tareas batch al desarrollo interactivo?

¿Son consistentes los estilos de interfases en todas las funciones?

¿Emplea la herramienta técnicas tales como el uso de botones, recuadros de diálogo y menús para evitar errores? Cuando ocurren errores ¿Envía la herramienta mensajes de error?

¿Ofrece la herramienta una modalidad experta (generalmente controlada por comandos)? Puede ser necesario buscar una herramienta que pueda manejar los diferentes niveles de conocimiento o calificaciones de sus desarrolladores.

¿Están disponibles abreviaturas para los comandos utilizados típicamente en modalidad experta?

¿Incluye la herramienta teclas funcionales de programa para acelerar la ejecución de las funciones? ¿Realiza asignaciones de tecla funcional estándar por ejemplo f1=ayuda? ¿Pueden asignarse dinámicamente funciones a teclas funcionales (por ejemplo macro instrucciones)?

¿Está disponible la ayuda a petición e indexada por temas?

¿Tiene la herramienta menús y/o barras de menú y son claras, bien organizadas y consistentes?

¿Se utilizan recuadros de diálogo para identificar opciones de selección sencilla o múltiple?

¿Puede el desarrollador abrir dos ventanas de forma que pueda referenciarse la información de una ventana mientras se trabaja en otra?

¿Podrían los desarrolladores de sistemas recorrer pantallas fácilmente para obtener la información que necesitan? Puede basarse en criterios tales como si existe una referencia cruzada para el uso de los datos o un mapa para guiar a los desarrolladores a través de las pantallas.

¿Pueden adaptarse las pantallas y otras funciones de herramientas a medida de las necesidades individuales del desarrollador?

¿Ofrece la herramienta algún comando como Undo que permita a los desarrolladores salir de una tarea fácilmente en cualquier punto? ¿Se acomoda la herramienta al trabajo en grupo?

¿Ofrece a varios desarrolladores acceso abierto a productos de trabajo en un repositorio CASE?

b) Definición de necesidades.

Es importante conocer los datos específicos de cómo la herramienta produce las definiciones de los procesos.

¿Ofrece referencias cruzadas para procesos y datos?

¿Las necesidades se introducen a las herramientas de manera automática?

¿La herramienta crea módulos reutilizables en los casos en los que las necesidades requieran rutinas comunes?

¿Puede generar casos para pruebas de amplio alcance en base a la necesidad?

¿Permite realizar pruebas de prototipo?

¿La herramienta CASE crea prototipos de pantallas y ventanas?

¿Ofrece algún soporte y puesta en práctica de metodologías y estándares?

c) Diseño.

Considerar el soporte ofrecido por las herramientas CASE para funciones clave de la fase de diseño:

- Diseño de pantallas.
- Diseño de reportes.
- Diseño de la base de datos.

¿Las interfases GUI son importantes para su empresa?

¿Soporta estándares GUI para títulos, barras de menú y asignaciones de teclas funcionales?

¿Permite la llamada y transferencia de parámetros?

¿Soporta las metodologías clave para su instalación?

¿Que sucede si existe una mejoría al sistema operativo de su empresa?, ¿responde adecuadamente la herramienta?

¿Permite que las especificaciones sean reutilizables?

¿Soporta diccionario de datos externos?

¿Soporta estándares para nombres de datos, archivos y módulos?

¿Permite utilizar técnicas de diagramas de flujo de datos?

¿Lleva a cabo la modelación de procesos?

d) Programación.

Genera código para se ejecutado en una variedad de sistemas operativos.

Aquí la persona que evalúa debe considerar las capacidades de programación que son factores críticos .

Por ejemplo:

- La versión del lenguaje generado debe ser compatible con sus versiones para compiladores, analizadores de código y herramientas de re-ingeniería.
- La herramienta debe soportar todos los comandos de programación del sistema administrador de bases de datos necesario.
- Al aparecer nuevas versiones del lenguaje, el vendedor debe mantener actualizada su herramienta.
- La herramienta CASE debe producir código estructurado.
- Debe generar nombres de programa, datos y subrutinas que resulten significativos.
- El código generado puede tener que incluir subrutinas no generales por el producto. Estas subrutinas generadas por la herramienta CASE deben ser utilizadas por programas externos que han sido programados utilizando otros medios.
- Deberán poder generarse y cambiar fácilmente modificaciones al lenguaje JCL ejecutable o cualquier otro lenguaje de comandos del sistema operativo necesario.
- La herramienta CASE deberá facilitar la reutilización de subrutinas comunes y estructuras de datos comunes.
- Debe producir código eficiente.

e) Pruebas.

La herramienta soporta las metodologías de pruebas existentes (alpha, beta, de integración).

Maneja test interactivos(permitiendo realizar cambios directamente a la aplicación).

f) Bases de datos.

En qué medida se ajusta la herramienta CASE a las capacidades del sistema administrador de bases de datos de la empresa.

Deberá tomar en cuenta las siguientes capacidades de bases de datos:

- El producto tiene que actuar en interfase con un diccionario de datos externo.
- El vendedor debe suministrar bases de datos y diccionarios internos fácilmente accesibles por otras herramientas.
- Manejar claves para acceder la base de datos.
- El producto debe soportar los estándares y nombres de datos de la empresa.
- La herramienta debe cumplir con los estándares de seguridad de la empresa.
- La herramienta CASE deberá ayudar en cierto modo a los desarrolladores a optimizar el rendimiento de las bases de datos.

g) *Runtime*.

Cuál es el rendimiento del código generado.

¿La herramienta CASE soporta los sistemas operativos y de ventanas apropiados para la empresa?

¿El vendedor ha realizado un buen trabajo en mejorar su producto para incorporar nuevas funciones y características de sistemas operativos?

¿Se requieren cambios a las bibliotecas *runtime* para el código generado?

¿Las aplicaciones tienen que ser generadas de nuevo para funcionar en diferentes sistemas operativos o sistemas de ventanas?

¿El código generado es portable?

¿El rendimiento del código generado puede ser adaptado para sistemas operativos específicos?

¿Existen restricciones en modalidad *runtime* respecto a la interfase con programas no escritos con el producto?

h) Entorno de proceso.

La herramienta está disponible para microcomputadoras, minicomputadoras y *mainframes*.

Considerando

¿Cuál es el límite de tamaño para un programa generado? ¿Existen limitaciones de tamaño para una aplicación?

Determine el número de usuarios que pueden utilizar la herramienta simultáneamente y verifique si

- ¿Se degrada el rendimiento al accederla más programadores?
- ¿El trabajo de cada desarrollador está protegido para no ser alterado por el trabajo del otro?
- ¿Cuánta memoria necesita?
- ¿Qué variables afectan al tiempo de respuesta de las herramientas que generan y comprueban codificación o de la aplicación generada?
- ¿Qué medios ofrece la herramienta para optimizar el rendimiento de la aplicación generada?
- ¿Qué espacio de almacenamiento necesita la herramienta para versiones anteriores y actuales de las aplicaciones generales?

i) Documentación.

Considerar si el tipo de documentación que ofrece es gráfica, está organizada, y si se puede llevar a cabo el mantenimiento de la documentación.

- ¿Soporta la herramienta gráficos tales como diagramas de flujo de datos, diagramas entidad-relación y diagramas de transición de estados?
- ¿Existe una capacidad *zoom* para enfocar un segmento o un gráfico?
- ¿Pueden codificarse en color componentes gráficos?
- ¿Puede la documentación generada cumplir con los estándares de la empresa?
- ¿Ofrece la herramienta un comando Undo que permita a los desarrolladores salir de una tarea fácilmente en cualquier punto? ¿Se acomoda la herramienta al trabajo en grupo?
- ¿Ofrece a varios desarrolladores acceso abierto a productos de trabajo en un repositorio CASE?

j) Administración del cambio.

Verificar si las versiones están sincronizadas en todos los niveles.

k) Análisis financiero.

Conocer su valor y ver si el costo queda compensado por las cualidades específicas de la herramienta CASE.

Hay que ver que el costo no sólo implicará el costo del producto, sino tarifas de licencia anual, costos de actualización, costos de asesoría, manuales, etc.

Por último la otra forma de seleccionar una herramienta CASE es seguir *la guía propuesta por Gibson* la cual hace referencia a: [GIB,88]:

- ¿Es la estructura fundamental de la herramienta CASE un DBMS o un diccionario de datos?
- ¿La herramienta CASE comparte o puede compartir en un futuro cercano formatos y especificaciones de archivos con otros productores de CASE?
- Describe la capacidad que tiene la herramienta CASE para leer procesos y librerías fuentes y crear especificaciones de componentes CASE para sistemas existentes.
- Describe como la herramienta CASE se comunica con otra herramienta de diseño CASE.
- ¿La herramienta tiene metodologías gráficas capaces de explotar los diagramas de diseño y especificaciones del diccionario a una profundidad razonable?
- ¿La herramienta es capaz de ejecutarse con capacidades Windows?
- Describe la capacidad que tiene la herramienta para realizar planeación estratégica en una

unidad funcional o en toda una empresa.

- ¿Provee un completo significado de prototipado?
- Describe la capacidad de la herramienta para proveer especificaciones de diseño físico a partir del diseño lógico.
- ¿Qué reportes de especificación de diseño la herramienta proporciona?
- ¿La herramienta provee un soporte analítico de la documentación del diseño?
- ¿Las especificaciones de diseño y desarrollo de la herramienta se comunican con un DBMS o un diccionario de datos?
- ¿Puede la herramienta ser modificada para una metodología de diseño existente en la organización?
- Describe la habilidad de la herramienta para generar diseño, operaciones y documentación del usuario.
- Describe la capacidad de la herramienta para conservar el diseño y cambiar el sistema.
- Describe la capacidad de la herramienta para generar programas para un rango de plataformas desde las especificaciones de diseño.

Como puede observarse, las dos guías para evaluar las herramientas tienen preguntas en común sin embargo, resulta ser más completa la publicada en computerworld.

4.8. CASE Y LA QUINTA GENERACION.

Sólo unos pocos años atrás se pensaba que las herramientas CASE mejorarían en aspectos de integración e inteligencia. La realidad es que las herramientas integradas están ya al alcance de todos y como se mencionó en capítulos anteriores, permiten automatizar todo el ciclo de vida del desarrollo de sistemas(I-CASE) .

Faltaría que las herramientas CASE además de ser potentes dejarán de estar limitadas. Gracias a la inteligencia artificial, aplicando tecnologías como el procesamiento de lenguajes naturales, reconocimiento de lenguaje natural, reconocimiento de la voz, ingeniería del conocimiento etc., es posible lograrlo.

Si las herramientas estuvieran rodeadas de un *shell* inteligente con sus tres componentes como :

- Interfaz inteligente,
- Sistema experto denominado " Conductor de la metodología",
- Librería de partes reutilizables,

permitiría aún más incrementar la calidad y la productividad en el proceso del desarrollo de software.

Explicando más a detalle estos tres componentes vemos que la interfaz de usuario que permite que el desarrollador de sistemas se enlace con la herramienta deberá de ser lo más amigable posible ya que determinará su uso posterior.

McClure señala que una interfaz que sea amigable se caracterizará por ser:

- Fácil de usar.
 - + Un número mínimo de conceptos a aprender serán necesarios para poder empezar.
 - + Un número mínimo de reglas de sintaxis.
 - + Un lenguaje unificado por todo el sistema.
- Robusto.
Sistema a prueba de usuario, respondiendo a todas las entradas del usuario, sean éstas correctas o incorrectas.
- Fácil de aprender.
No se requiere de muchas reglas de sintaxis ni muchos conceptos cualquier usuario podrá realizar consultas.
- Adaptable.
El sistema se ajustará al usuario experto y será diferente a como lo hará con el usuario inexperto.
- Colaborador.
Los comandos de ayuda estarán siempre disponibles.
- Detecta e informa sobre los errores.
Avisará si se ha cometido un error, explicando cómo corregirlo.

Cabe aclarar que no sólo se buscará que la interfase sea amigable sino orientada al usuario cumpliendo las características:

- Simple.
- Consistente.
- Flexible.
- Controlada por el usuario.
- Lenguaje natural.

Ahora bien si el sistema es amigable y orientado al usuario deberá ser reactivo, respondiendo a las peticiones del usuario y aprenderá y estudiará cómo el usuario utiliza el sistema. Esto es con el fin de que el sistema reactivo pueda cambiarse fácilmente para satisfacer mejor las necesidades del usuario.

Lo que se busca con este tipo de sistemas reactivos es proporcionar un consejo constructivo cuando el usuario ha cometido un error, y quizás corregirlo por él. Pudiendo el sistema analizar la entrada y de acuerdo a sus sospechas cambiará la entrada, informando al usuario de la corrección, si la corrección no era la correcta, el sistema permitirá que el usuario cancele la corrección.

Ayudado por una opción de *help* podrá solucionar todos los problemas que pudieran presentarse durante la operación del sistema.

Si el sistema llegase a incorporar ya no un simple *help* sino un tutor inteligente serán mucho mejor que tener a un instructor humano. Ello permitirá que el usuario aprenda más sobre algo en particular e inclusive le puede proponer solución a sus problemas.

Este tutor estará basado en un profesor experto que utilice el lenguaje natural para comunicarse, y no sólo enseñará sino que aprenderá del usuario.

El otro componente que se mencionó sería el conductor de la metodología, indispensable para aquellos que son todavía inexpertos en el desarrollo de software, permitiendo que se concentren más en la estrategia que en los aspectos del proceso.

En realidad este conductor de la metodología será un sistema experto, basado en profesionales expertos del desarrollo de software.

Con sus componentes básicos : Herramientas de software integradas, estrategias y heurísticas del desarrollo de software, reglas y hechos de la tecnología de software y principios básicos, formando estos últimos tres la base de conocimientos, guiarán la utilización del conjunto de herramientas integradas.



Fig. 4.8 Componentes del conductor de la metodología

Empleando técnicas de inteligencia artificial como la lógica difusa, redundancia, razonamiento probable y revisión de las ideas el conductor de la metodología podrá trabajar con datos incompletos e incorrectos transformándolos a un sistema completo.

Para terminar con los componentes del *shell* mencionaremos que si deseamos utilizar algún componente del software lograremos realmente incrementar la productividad. Para ello necesitaremos de un depósito CASE en el cual estén almacenados aquellos componentes que puedan ser reutilizables como:

- Prototipos reusables.
- Datos reutilizables.
- Arquitecturas reutilizables de sistemas y programas.
- Diseños reutilizables de programas y estructuras de datos.
- Modelos reutilizables de datos.
- Programas de códigos reutilizables.
- Paquetes de software reutilizables.

Este depósito CASE será el mecanismo que proporcione la librería de partes reutilizables y el sistema de administración de la librería para almacenar y recuperar los componentes y facilitar el acceso y la búsqueda, el control de la versiones y la seguridad. De esta manera se logrará reducir el tiempo de desarrollo y el costo, mejorando la calidad e incrementando la productividad.

4.9. CICLO DE VIDA TRADICIONAL CONTRA EL CICLO DE VIDA CASE.

Como se mencionó en el capítulo 1 de este proyecto la metodología del ciclo de vida de desarrollo de un sistema consta de múltiples fases que van desde una planeación del sistema hasta el mantenimiento del mismo, incluyendo herramientas, técnicas y métodos que guían y controlan todas las actividades que se llevan a cabo en cada fase.

Con el enfoque tradicional de desarrollo de sistemas(secuencial) se dedica aproximadamente un 35 % de los recursos al análisis y diseño, dando como resultado sistemas mal especificados y de alto costo.

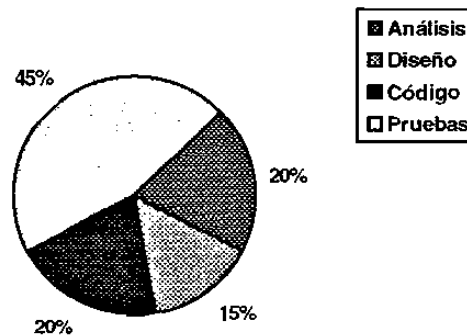


Fig. 4.9 Ciclo de vida tradicional del sistema

Gracias a las técnicas estructuradas se logró dar mayor énfasis a las primeras fases en el ciclo de vida de desarrollo de sistemas.

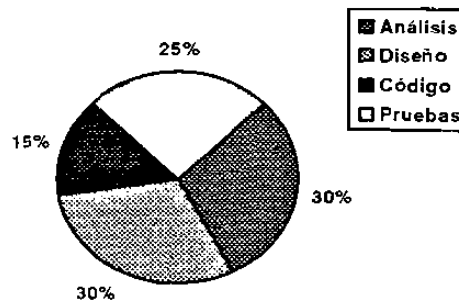


Fig. 4.10 Ciclo de vida del sistema con técnicas estructuradas

Pero el cambio realmente fue drástico con CASE, ya que el énfasis hacia el análisis y diseño es mayor logrando así una alta productividad y un software con menos errores.

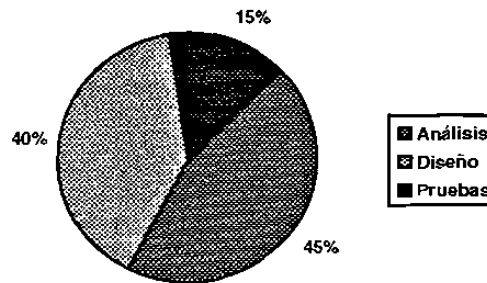


Fig. 4.11 Ciclo de vida CASE del sistema

Si se está acostumbrado a utilizar una metodología de desarrollo tradicional, podrá encontrar la gran diferencia al utilizar CASE:

DESARROLLO TRADICIONAL	CON CASE
<ul style="list-style-type: none">- Énfasis en el desarrollo tradicional.- Énfasis en la codificación y en las pruebas.- Codificación manual.- Documentación manual.- Pruebas de software.- Mantenimiento de código.	<ul style="list-style-type: none">- Énfasis en el análisis y diseño (estructurado).- Manejo de la técnica de prototipo.- Generación automática de código.- Generación automática de la documentación.- Comprobación automática del diseño.- Mantenimiento de las especificaciones del diseño.

Tabla 4.2 Desarrollo tradicional vs. desarrollo con CASE

Mucho se ha hablado de lo que implica desarrollar un software; de la calidad total que se requiere en los productos que se fabrican, y de la nueva tecnología CASE que automatiza el proceso de desarrollo de sistemas.

Pero todo esto qué relación tendrá finalmente si el objetivo de nuestro estudio es buscar la manera de guiar a los desarrolladores de sistemas a través de estrategias para lograr un sistema o software con calidad. Simplemente es importante recordar y considerar que la crisis del software que se ha venido presentando aquí en nuestro país representa no sólo un peligro sino una oportunidad de innovar y salir adelante en las perspectivas de desarrollo de software con calidad. Ello se hará basándonos en la nueva tecnología CASE y sin olvidarnos, por supuesto, del desarrollo de una nueva cultura a través de un proceso de cambio guiado por una administración de calidad y soportado por capacitación y educación en nuevos métodos, técnicas y herramientas. Esto da como resultado personas con calidad que conlleven a desencadenar el proceso de desarrollo de calidad en el área de desarrollo de sistemas en nuestra empresa.

De esta manera si queremos que un sistema tenga calidad, estamos seguros que CASE como componente de los sistemas técnicos y los sistemas sociales o bien la cultura, agregarán ese valor al sistema logrando así la calidad de los procesos dirigida a la calidad del producto para así producir la información requerida por los usuarios en el tiempo previsto.

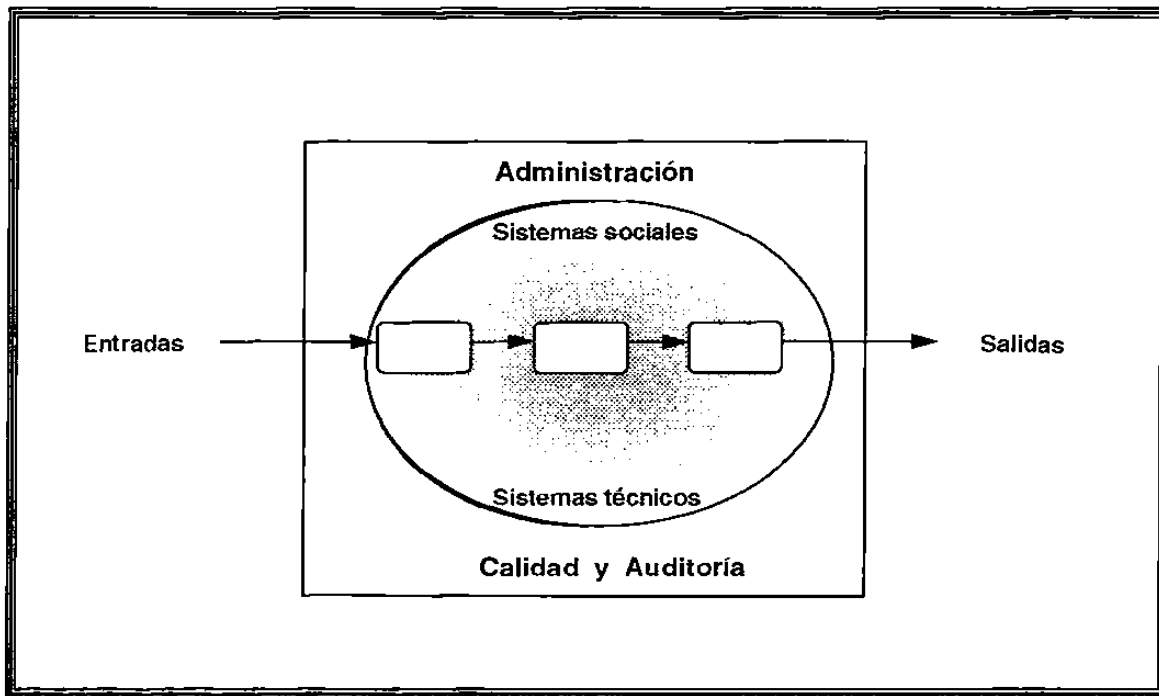


Fig. 4.13 Elementos a considerar para lograr la calidad en un sistema

4.10. CASE EN ALGUNAS EMPRESAS DE MONTERREY.

Aplicando el cuestionario número uno del anexo B para determinar el uso que se le da a la tecnología CASE pudo observarse que son muy pocas las empresas que realmente la utilizan en el desarrollo de sus proyectos, ya que en otros casos las tienen pero no saben cómo utilizarlas por el hecho de que no dominan las técnicas que soporta la herramienta.

Se mencionarán algunas de las herramientas que están siendo adquiridas por algunas de las empresas, las cuales se han clasificado por el tipo de industria a la que pertenecen. Lo que podemos comprobar es que aquéllas que tienen ya aproximadamente un año de utilizarlas, aseguran mejoras en calidad y productividad.

INDUSTRIA VIDRIO	PLATAFORMA AS/400 IBM		
HERRAMIENTA AS/SET(AS/400 Software Engineering Technology)	CATEGORIA LOWER CASE		
<p>GENERALIDADES:</p> <ul style="list-style-type: none"> • Permite realizar la modelación de datos, prototipo y generación de código. • El código que genera es en RPG/400 nativo. • No requiere de run-time, siendo los programas portables a otras AS/400 sin que AS/SET esté residente en el sistema. • Su menú principal está formado por las opciones de (Display, Report Programs, Batch Programs, Repository, Product Security). • El papel del analista como del administrador de bases de datos en la empresa al usar este tipo de herramientas es: <table style="width: 100%; border: none;"> <tr> <td style="text-align: center; width: 50%;"> <p>Analista</p> <ul style="list-style-type: none"> - Elaborar diccionario de datos. - Diseñar archivos. - Modelación de datos. - Programación. </td> <td style="text-align: center; width: 50%;"> <p>Administrador de Base de datos</p> <ul style="list-style-type: none"> - Introducir diccionario de datos al sistema. - Armar archivos en AS/SET. - Copiar modelos de datos al SET de aplicaciones de cada analista. </td> </tr> </table>		<p>Analista</p> <ul style="list-style-type: none"> - Elaborar diccionario de datos. - Diseñar archivos. - Modelación de datos. - Programación. 	<p>Administrador de Base de datos</p> <ul style="list-style-type: none"> - Introducir diccionario de datos al sistema. - Armar archivos en AS/SET. - Copiar modelos de datos al SET de aplicaciones de cada analista.
<p>Analista</p> <ul style="list-style-type: none"> - Elaborar diccionario de datos. - Diseñar archivos. - Modelación de datos. - Programación. 	<p>Administrador de Base de datos</p> <ul style="list-style-type: none"> - Introducir diccionario de datos al sistema. - Armar archivos en AS/SET. - Copiar modelos de datos al SET de aplicaciones de cada analista. 		

Tabla 4.3. CASE en empresas que pertenecen a la industria del vidrio

INDUSTRIA CELULOSA Y DERIVADOS	PLATAFORMA MACINTOSH CLASSIC
HERRAMIENTA ANATOOL	CATEGORIA MIDDLE CASE(Parcial)
<p>CARACTERISTICAS:</p> <ul style="list-style-type: none"> • Soporta el análisis estructurado. • Genera la documentación del análisis. • No soporta la administración de proyectos , prototípeo , y generación de código. • Puede exportar diccionario de datos como archivo ASCII. • Distribuidor: Advanced Logical Software 9903 Santa Monica Blvd. Beverly Hills, CA 90212. 	

Tabla 4.4 CASE en empresas que pertenecen a la industria de celulosa y derivados

INDUSTRIA Productos alimenticios	PLATAFORMA PC386
HERRAMIENTA POSE Versión 4.11	CATEGORIA UPPER CASE, MIDDLE CASE
<p>GENERALIDADES:</p> <ul style="list-style-type: none"> • Esta herrmamiento es distribuida por CSA(Computer Systems Advisors). • Cuenta con un repositorio que almacena todos los objetos que se definen durante el proceso de desarrollo. • Herramienta Modular. • Siendo los módulos con los que cuenta esta herramienta: <ul style="list-style-type: none"> - Diagramador de modelo de datos. - Normalizador de modelo de datos. - Diseñador de base de datos lógica. - Ayuda en bases de datos. - Prototípeo de pantallas y reportes. - Diagramador de flujo de datos. - Diagramador de estructuras. - Diagramador de descomposición. - Diagramador de acciones. - Diagramador de matriz de planeación. 	

INDUSTRIA Productos alimenticios	PLATAFORMA AS/400 IBM
HERRAMIENTA AS/SET, SYNON, ADELIA	CATEGORIA I-CASE
<p>GENERALIDADES (ADELIA) y comentarios :</p> <ul style="list-style-type: none"> • Esta es la herramienta más utilizada en la empresa. • Herramienta construida en lenguaje C para OS/2. • Sólo se requiere de tres días de entrenamiento. • Permite mejorar la comunicación entre el desarrollador y los usuarios con el prototípeo. • Contiene un repositorio que maneja toda la información del proyecto. • Puede trabajar en micro o en red a través de una transferencia de información del repositorio a la micro sólo que los programas deberán de ser compilados en AS/400. • Su costo es de \$25,000 dolares. • La versión utilizada es 6.0. • Es muy amigable. • Genera código en RPG/400 y COBOL/400 siendo las características de los programas generados: modularidad, estructurados y de tamaño adecuado. • El soporte técnico de esta herramienta es dado por la compañía STR de Monterrey N.L. • La metodología que utiliza la empresa y que se vende con la herramienta es Merise. 	

Tablas 4.5 CASE en empresas que pertenecen a la industria de productos alimenticios

EMPRESA Institución Financiera	PLATAFORMA PC's, Mainframe: IBM, UNISYS, BULL, DEC/VAX
HERRAMIENTA PacBase	CATEGORIA I-CASE
<p>GENERALIDADES :</p> <ul style="list-style-type: none"> • Soporta la metodología de Yourdon, Merise, SSADM (Structured systems Analysis and Design Method). • El costo para mainframe de cada módulo que se seleccione es de \$150,000 a \$450,000 dolares. • Versión que utilizan 8.0.2. • Es una herramienta I-CASE que soporta a todas las fases del ciclo de vida del desarrollo de un sistema, control de calidad además de contar un módulo que soporta la administración del sistema. • Genera programas en COBOL. • Las especificaciones que se definen para un sistema pueden ser reusables. 	

Tabla 4.6 CASE en instituciones financieras

INDUSTRIA Maquinaria e implementos agrícolas	PLATAFORMA AS/400- ADELIA, SilverRun- PC's
HERRAMIENTA ADELIA, SilverRun	CATEGORIA ADELIA-I-CASE, SilverRun- Middle CASE
<p>GENERALIDADES :</p> <ul style="list-style-type: none"> • Esta herramienta soporta a la metodología de Merise, soportando SilverRun a otras como Gane-Sarson y Yourdon-DeMarco. • La herramienta SilverRun la utilizan principalmente para desarrollar el modelo conceptual y relacional del sistema, ADELIA se utiliza para construir la aplicación o el sistema. • Para ADELIA se requiere de 100 Mb de disco duro, 8 Mb de memoria. • El costo de ADELIA para la empresa fue de \$25,000 dolares. • Cada módulo de SilverRun fué de \$2,500 dolares con un traductor para la AS/400 de \$1500 dolares dando un total de \$11,500 dolares. • La versión con la que cuenta la compañía es ADELIA versión 6 y SilverRun versión 2.11. • ADELIA cuenta con un módulo para la administración de proyectos. • Una de las ventajas de la herramienta ADELIA no sólo es la generación de código sino que cuenta con un 4GL para desarrollo de aplicaciones. • El soporte técnico para ambas herramientas es proporcionado por STR en Monterrey. 	

Tabla 4.7 CASE en empresas que pertenecen a la industria de maquinaria e implementos agrícolas.

Como se puede observar predomina mucho el uso de herramientas como ADELIA y SilverRun ya que uno de su distribuidores se encuentra aquí en Monterrey y es la compañía llamada:

STR
Compañía de Software
Loma Panorámica 361-7.
Col. Loma Larga.
Monterrey, N.L. CP.64710.
Tel. 344-74-12.

esta compañía después de una presentación de la herramienta ofrecerá prestarle la herramienta(software y sus manuales) por un mes para su evaluación, sin ningún compromiso durante un mes, tiempo razonable si tiene la gente capacitada en las técnicas estructuradas y en métodos o guías de evaluación.

Quizá porque todavía no se conozcan más compañías que distribuyen herramientas CASE en Monterrey o en nuestro país, las compañías tienen miedo de comprar herramientas en E.U.A. porque quizás éstas no ofrecen el soporte o la capacitación necesaria aquí en México.

Es por eso que aquí mencionamos otras opciones por si se desea evaluar para después adquirir una herramienta CASE.

FACTUM, S.A. de C.V.
J. Luis Lagrange No. 103-8° Piso.
Col. Los morales Polanco.
11510-México-D.F.
Tel. 557-94-18
Fax.557-93-69

Distribuidor Exclusivo de KnowledgeWare, Inc.

Sus herramientas : **ADW 2.7 (Modular).**
Legacy Workbench.
CWS/GUI.
Flashpoint.
Object View.

Herramienta: Prokit * Workbench.
Representante de ventas en México.
PROTECSA.
Julio Verne #11 Colonia Polanco.
México D.F. 11560.
Tel. 525-16-27.

Herramienta: IEW.
Representante Mancera Hermanos y Cía. S.C.
Jaime Balmes #11 Torre "D" Piso 4.
Plaza Polanco Colonia los Morales.

Polanco México D.F. 11510.
Tel. 556-91-73.

Herramienta: DesignAid
Representante de ventas en México.
Despacho Roberto Casas Alatraste.
Malamoros Pte. 1441.
Monterrey N.L.
Tel. 3-40-56-69 y 344-97-84

4.11. PERFIL DE LAS EMPRESAS EN MONTERREY Y SU AREA METROPOLITANA.

Se decidió aplicar un cuestionario para obtener el perfil de las empresas que están desarrollando sistemas de información, y conocer tanto el tipo de tecnologías que soportan su desarrollo de sistemas como los problemas a los que se enfrentan continuamente durante este desarrollo, y que causan el incumplimiento de los planes y los requerimientos. Dicho cuestionario es la fuente principal para la recopilación de información de diversas fuentes y se ha dividido en seis secciones:

1. Perfil de la compañía/organización.
2. Antecedentes del departamento de sistemas.
3. Uso de metodologías.
4. Problemática en el desarrollo de sistemas.
5. Experiencia en CASE.
6. Puntos de vista del encuestado.

Fué aplicado a una muestra de 53 empresas, y fueron contestados sólo 17 cuestionarios por personas cuya función en el departamento de desarrollo de sistemas, era de líder de proyectos o analista en la mayoría de los casos; el resto de las empresas que no contestaron justificaban el no poderlo hacer por las siguientes razones:

- Es información estratégica de la empresa.
- No hay tiempo, ya que hay exceso de trabajo.
- Es posible dar respuesta en dos o tres meses.
- El único que puede contestarla viaja continuamente.
- Despiden a gente por proporcionar información.

Sin embargo a pesar de que la muestra se redujo, el resultado de la recopilación de información será una explotación de la misma en forma satisfactoria.

La muestra resultante la constituyen empresas pertenecientes a diversos tipos de industria, así como empresas dedicadas a comercializar diferentes tipos de productos, entre ellas se encuentran:

- Empresas que pertenecen a la industria :

- + Productos alimenticios. (1)
- + Cerveza y Malta. (2)
- + Tabaco. (1)
- + Química. (2)
- + Fibras artificiales o sintéticas. (1)
- + Construcción. (2)
- + Servicios de análisis de sistemas y procesamiento de información. (1)
- + Servicios de asesoría y estudios técnicos de ingeniería y arquitectura. (1)
- + Acero. (1)
- + Metal mecánica. (1)
- + Comunicación telefónica. (1)

- Empresas dedicadas a la comercialización. (3)

Resultados:

A continuación los resultados obtenidos serán presentados en forma gráfica y explicados brevemente:

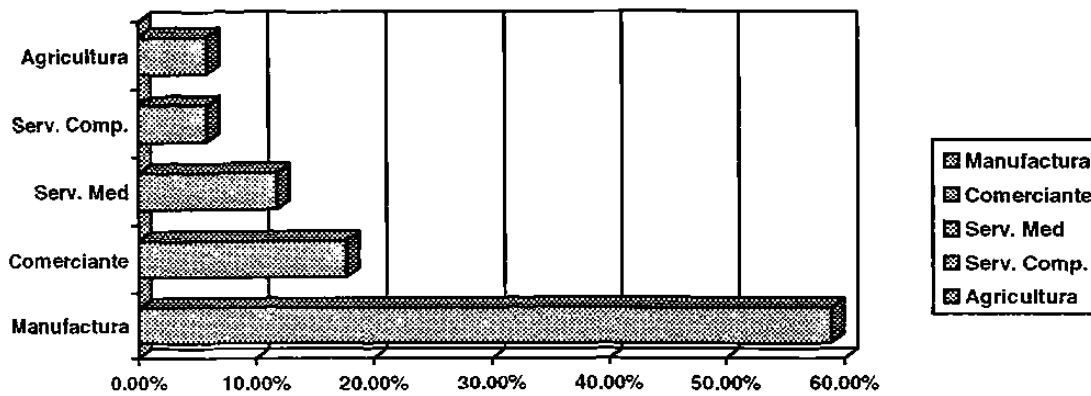


Fig.4.13 Tipos de industria

Monterrey siempre se ha distinguido por ser una ciudad industrial particularmente en el área de manufactura es por eso que en esta gráfica el porcentaje mayor corresponde al giro de manufactura, sin embargo debido a que la informática es parte ya de cualquier tipo de empresa se puede observar su participación de otros giros en este estudio.

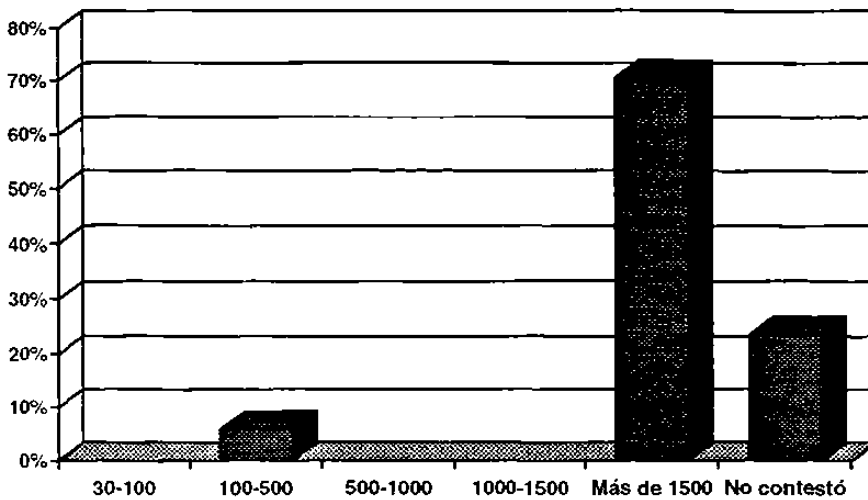


Fig.4.14 Ingresos brutos por año (en miles de nuevos pesos).

La mayoría de las compañías (70.58%) tienen ingresos brutos de más de 1500 nuevos pesos anuales.

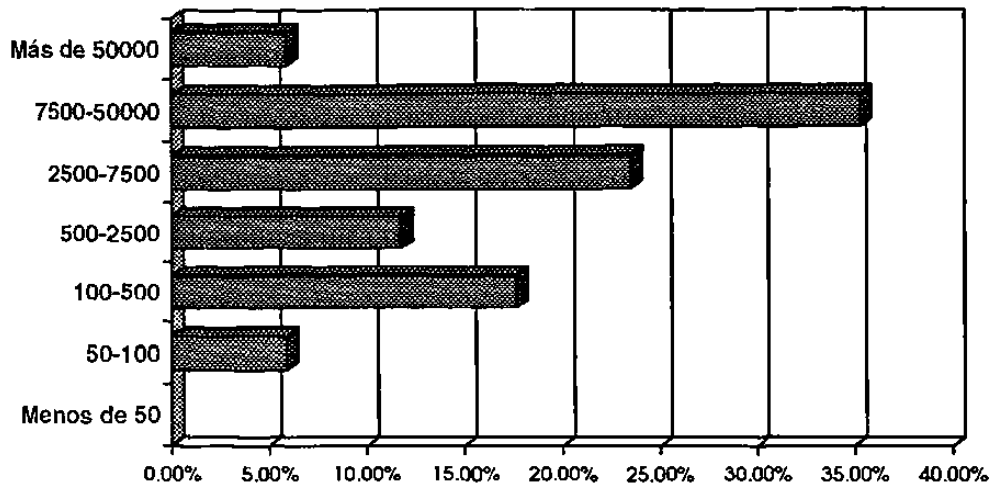
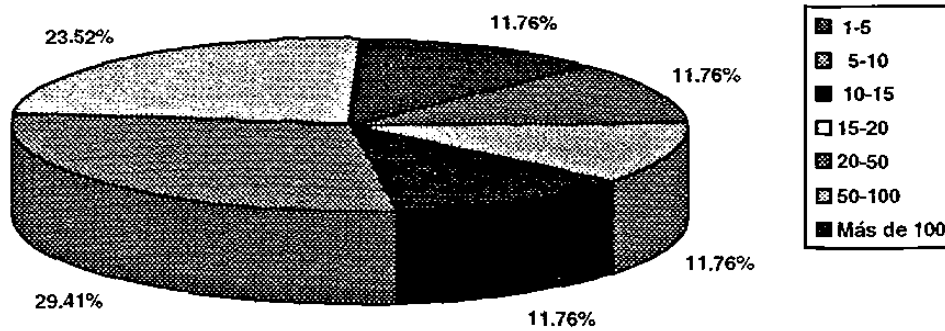


Fig.4.15 Cantidad de empleados en la empresa

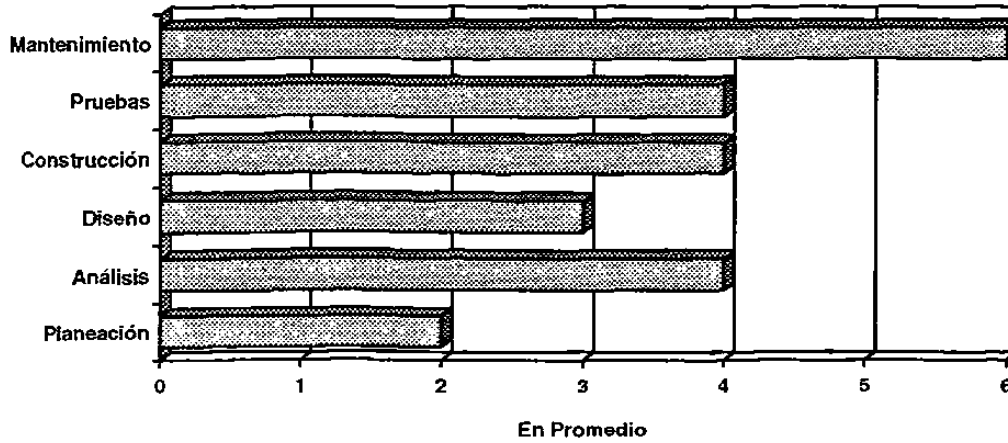
Asimismo, la mayoría de las empresas son grandes ya que el 23.57% tiene de 2500-7500 empleados, el 35.25% tiene 7500-50000 empleados y el 5.88% tiene más de 50,000 empleados.

Fig.4.16 Cantidad de empleados en el departamento de informática



En el departamento de sistemas o informática de muchas empresas existe gran cantidad de personas, sin embargo en la gráfica siguiente veremos que la mayoría de las personas están destinadas a actividades de "mantenimiento".

Fig.4.17 Cantidad de empleados destinados al desarrollo de sistemas



Por medio de esta gráfica nos damos cuenta de que uno de los mayores problemas en el área de desarrollo de sistemas es la asignación de recursos tanto humanos como materiales al mantenimiento de los sistemas ya existentes dejando a un lado el desarrollo de nuevos sistemas. Esto es un verdadero problema ya que el costo de mantenimiento implica una reducción en la productividad.

La causa de un mantenimiento excesivo en los sistemas implica la falta de metodologías, técnicas, herramientas ; una adecuada administración del desarrollo de sistemas, la falta de un involucramiento total del usuario así como un departamento con gente falta de cultura computacional.

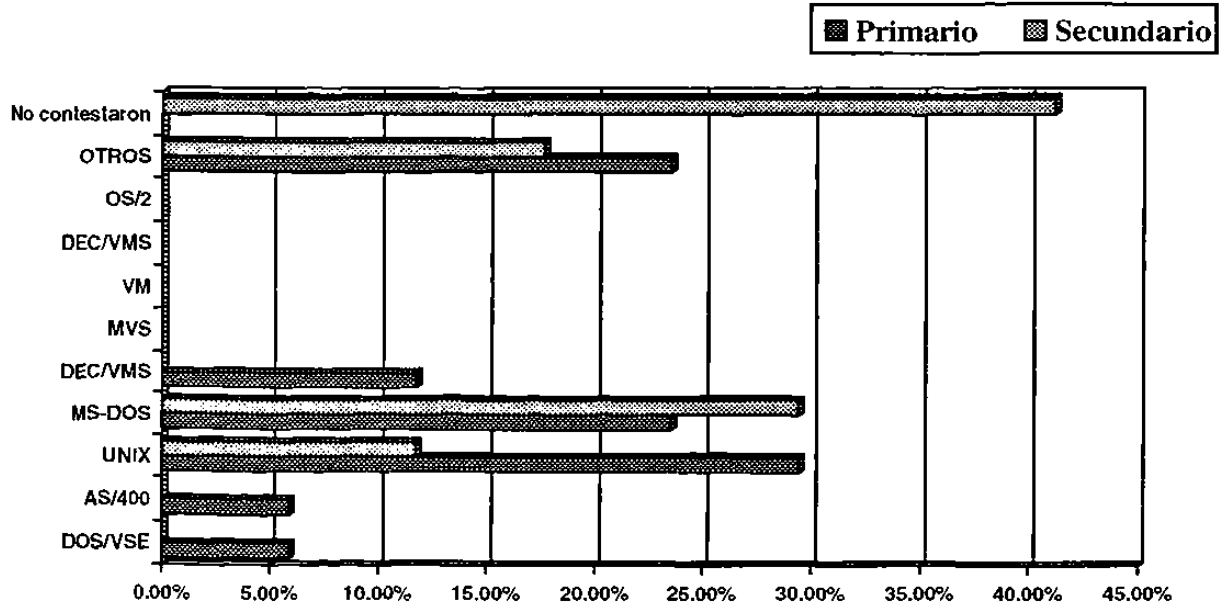


Fig.4.18 El sistema operativo para el medio ambiente de producción

Esta gráfica al igual que la siguiente, muestran cómo el MS-DOS y el UNIX constituyen la plataforma de desarrollo y producción para los sistemas en las empresas.

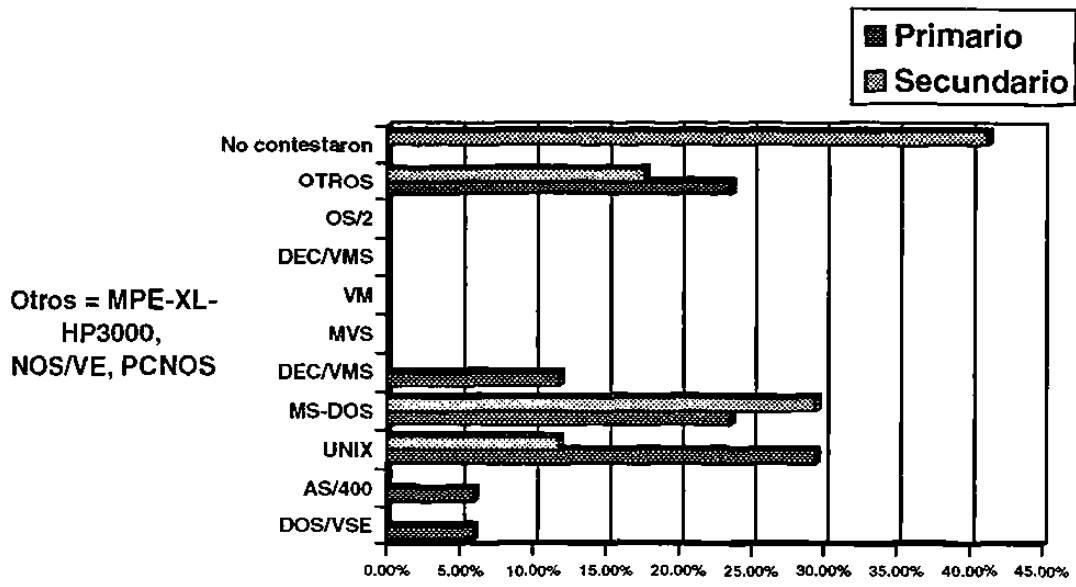


Fig.4.19 El sistema operativo paora el medio ambiente de desarrollo

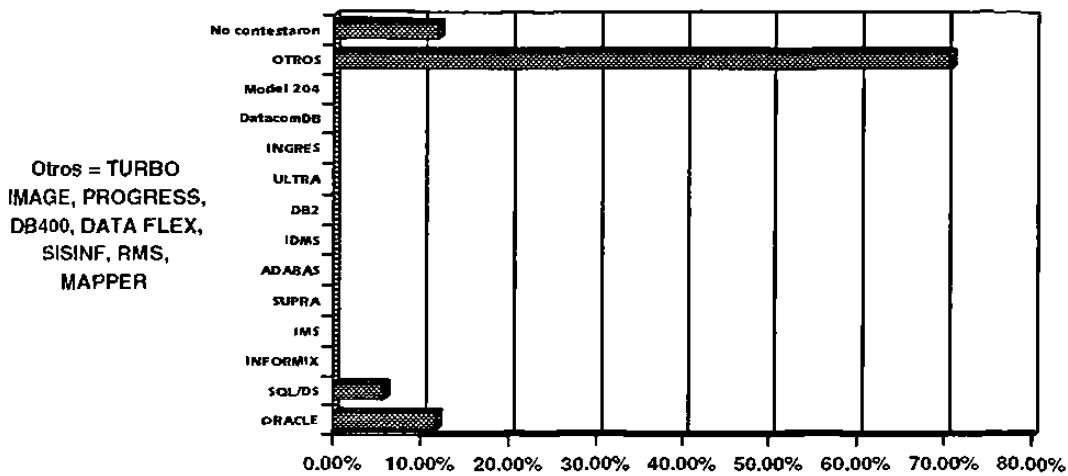


Fig.4.20 Principales sistemas administradores de bases de datos (DBMS)

En esta gráfica vemos que los administradores de Bases de Datos clasificados en la categoría de "otros" representan los sistemas más utilizados en las empresas, siendo PROGRESS el de mayor uso con un (23.52%) del 70.5 %

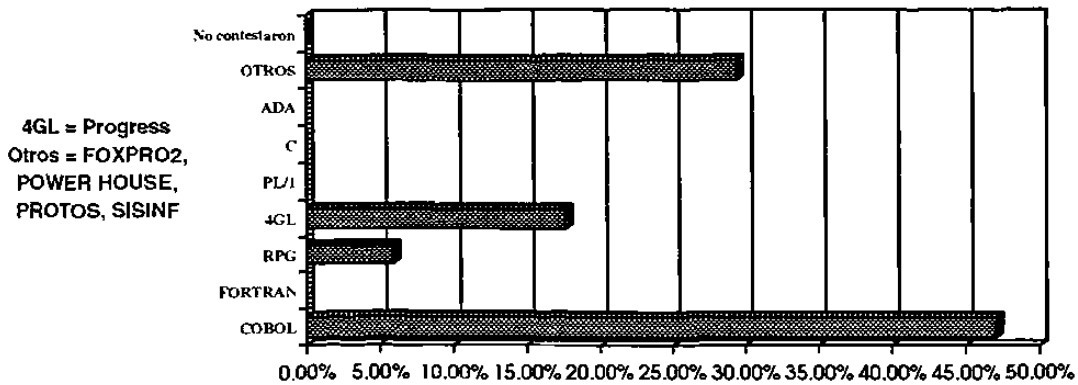


Fig.4.21 Principales lenguajes de desarrollo

Actualmente, las empresas empiezan a moverse a lenguajes más poderosos para soportar el desarrollo de sistemas, siendo estos lenguajes los denominados lenguajes de cuarta generación, sin embargo en la mayoría de ellas el COBOL sigue siendo el lenguaje base. Aunque en algunas empresas utilizan lenguajes de cuarta generación con metodologías de tercera generación.

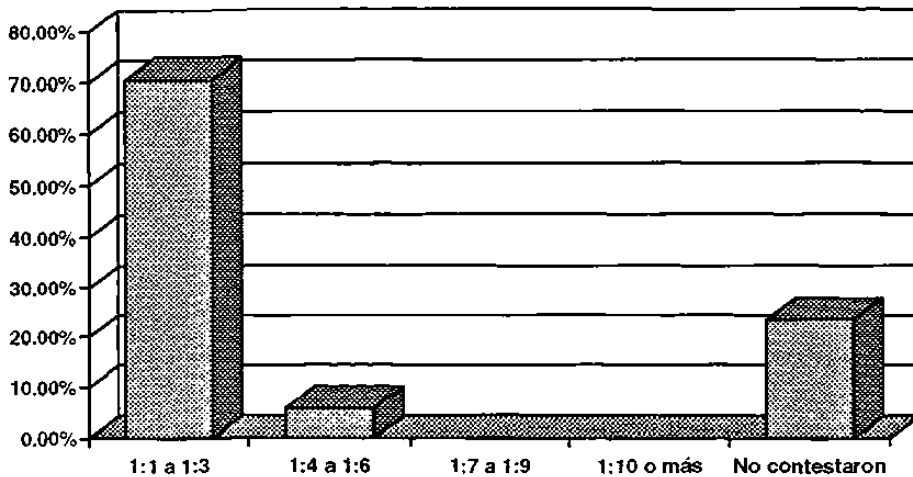


Fig. 4.22 Proporción de PC's, estaciones de trabajo o terminales para los desarrolladores

Podemos ver que el área de desarrollo de sistemas en estas empresas cuenta con suficientes recursos(hardware) para su persona, tal que no sería esto un pretexto para no terminar conforme a lo planeado un sistema requerido. No se puede atribuir este factor al problema de la inproductividad.

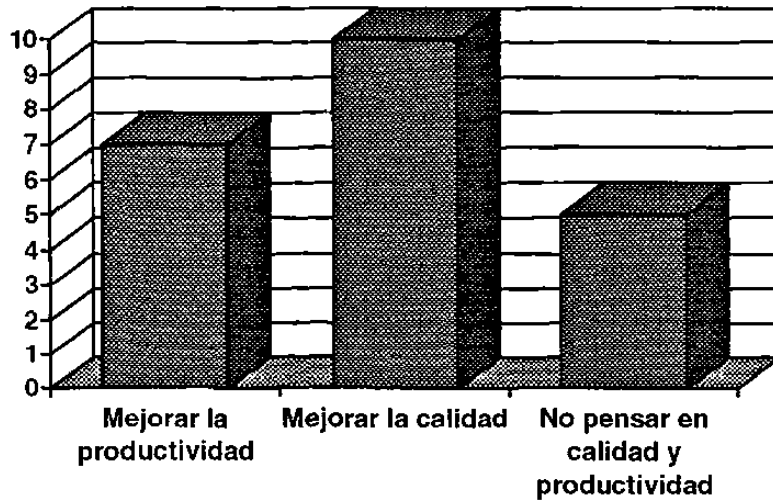


Fig.4.23 Lo más importante para la organización

Debido al medio ambiente competitivo al que se enfrentan las empresas, deberán emprender un proceso de mejora continua y constante de calidad, aquí observamos que de las 17 empresas 10 consideran de mayor prioridad mejorar la calidad de sus productos o servicios.

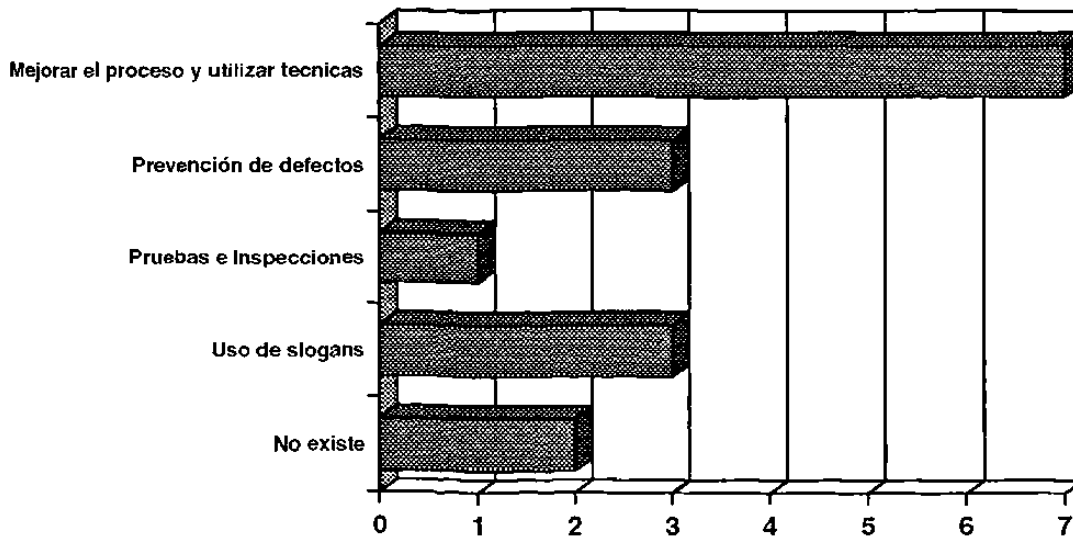
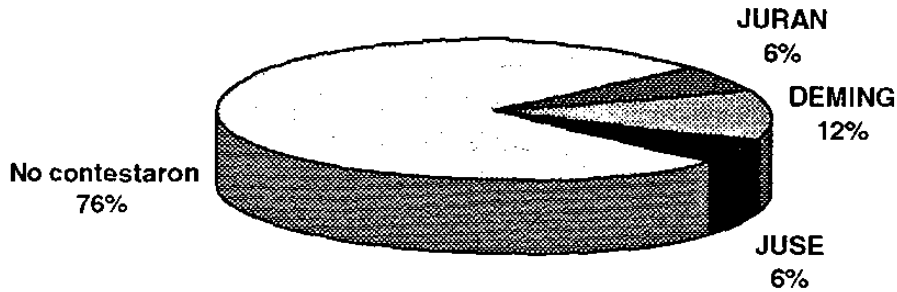


Fig.4.24 Características de un programa de calidad utilizado en las empresas

Debido a que las empresas se encuentran en un proceso de calidad éstas utilizan programas y técnicas de calidad, siendo las más utilizadas las técnicas estadísticas para control de procesos, pudiendo ser quizás éstas utilizadas para controlar el proceso de desarrollo de sistemas.

Fig.4.25 Filosofía de calidad que las empresas utilizan en el desarrollo de sistemas



Son muy pocas las empresas que toman como base una filosofía de calidad para soportar el proceso de desarrollo de sistemas, de hecho las filosofías que estas empresas utilizan no están adaptadas al departamento de informática.

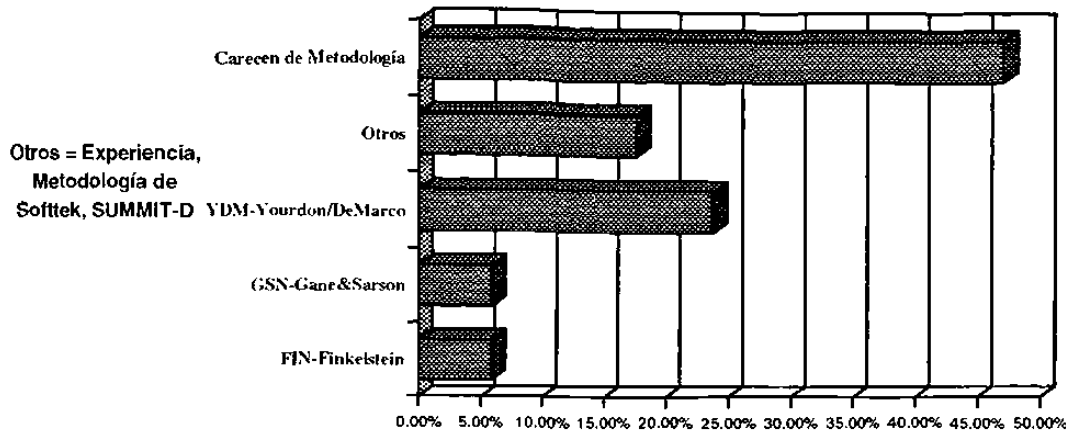


Fig.4.26 Uso de metodologías para el desarrollo de sistemas
Métodos estructurados

Como se podrá observar las metodologías que son utilizadas en el área de desarrollo de sistemas son basadas en la experiencia y muchas de las metodologías de reconocidos autores no se siguen al pie de la letra, es por esto y por la carencia de una metodología ya sea formal o desarrollada en la empresa que los sistemas no llegan a terminarse dentro del tiempo y presupuesto previstos además de no cumplir con los requerimientos de calidad previstos.

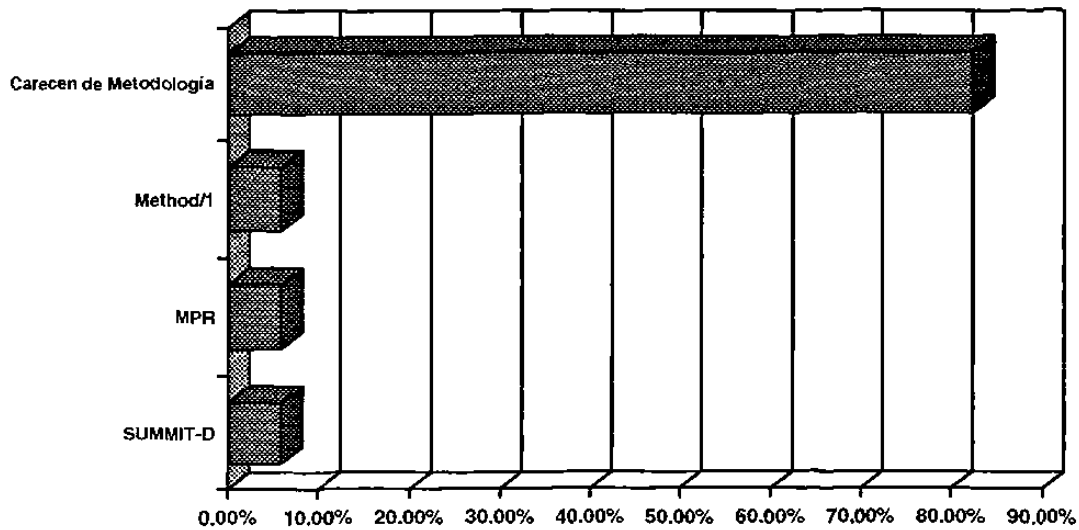


Fig.4.27 Uso de metodologías para el desarrollo de sistemas
Metodologías del ciclo de vida

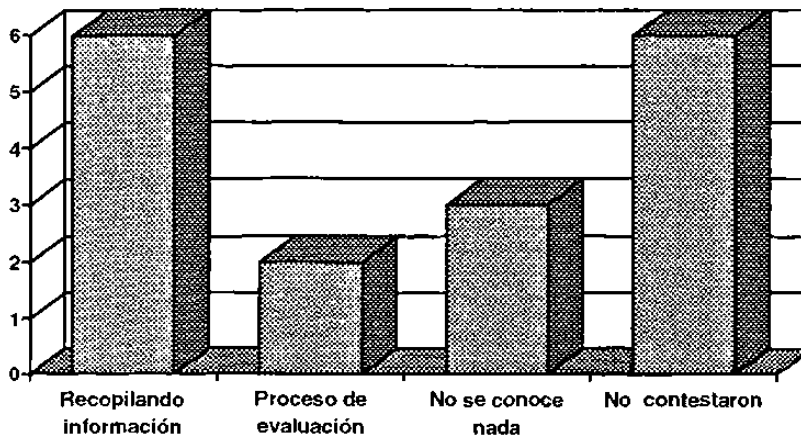


Fig.4.28. Experiencia en relación a las herramientas CASE

La mayoría de las empresas no conocen las herramientas CASE, son sólo pocas las que están comenzando a investigar su significado y la manera como CASE puede beneficiar al área del desarrollo de sistemas. Dado que la cultura computacional es muy pobre en algunas de estas empresas será necesario que antes de conocer o investigar o evaluar una herramienta CASE se capaciten en metodologías y técnicas, pudiendo posteriormente elegir la herramienta más adecuada para soportar el desarrollo de sistemas, tomando como base las metodologías del departamento.

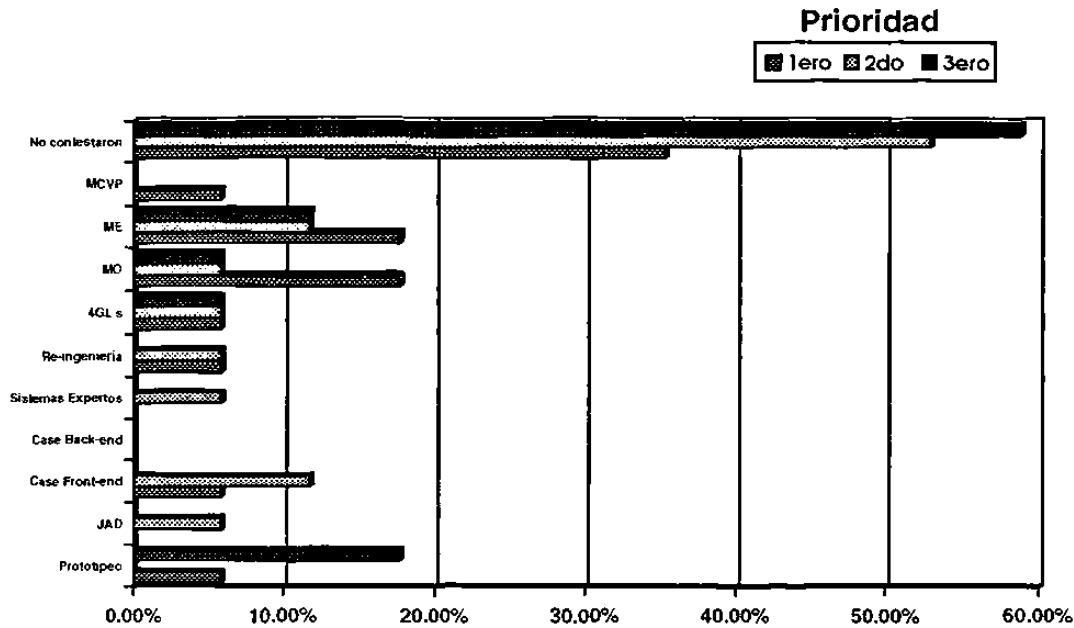


Fig.4.29 Enfoques para mejorar el proceso de desarrollo en la organización

Si nos ponemos analizar los resultados obtenidos por esta pregunta veremos que tanto la metodología orientada a objetos (MO) y los métodos estructurados (ME) se consideran como primordiales para mejorar el proceso de desarrollo de sistemas en la empresa, podríamos pensar que (MO) no dominada por muchos y los métodos estructurados sólo por algunos son indispensables el hecho es que el departamento no deberá de pensar que la sola metodología contribuirá a una mejora relevante, es decir se requiere de una combinación de todos los elementos arriba mencionados los cuales permitirán que se de un valor agregado al proceso del desarrollo mejorándolo y contribuyendo a lograr un producto de calidad.

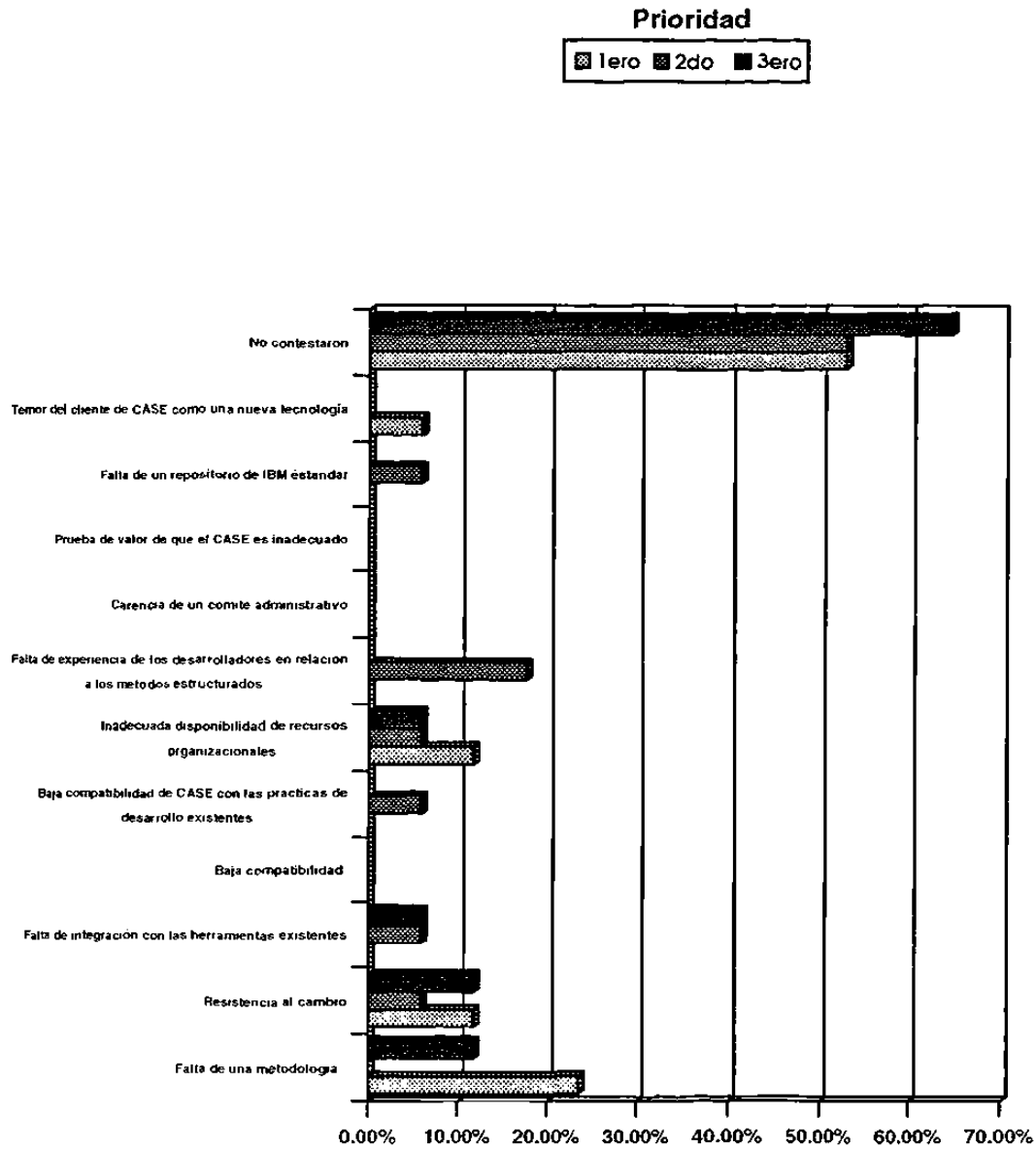
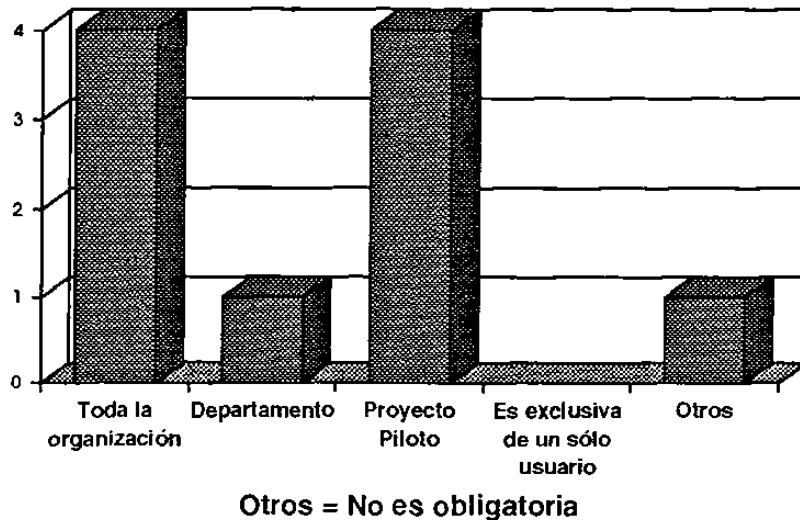


Fig.4.30 Principales obstáculos para adoptar CASE en la organización

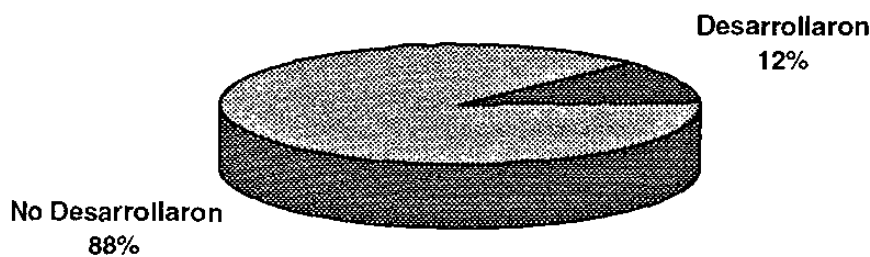
Si analizamos la gráfica anterior podemos ver que a pesar de que las personas han opinado qué se requiere para mejorar de metodologías(orientadas a objetos o estructuradas), su inexperiencia en ellas y la falta de las mismas, no sólo contribuirán a seguir con la crisis de software por todos conocida, sino que no permitirán que una nueva tecnología como CASE se institucionalice en la empresa.

Fig.4.31 Recomendación de las empresas para la implementación de CASE



En dado caso que se llegara a investigar y evaluar herramientas CASE decidiéndose por alguna de ellas que cumpla las expectativas y que contribuya al logro de los objetivos del departamento, es quizás recomendable aplicarla al desarrollo de todos los nuevos sistemas en toda la organización o aplicarla a un proyecto piloto. La segunda opción es la que mejor recordamos por los diversos enfoques y modelos para implementar CASE.

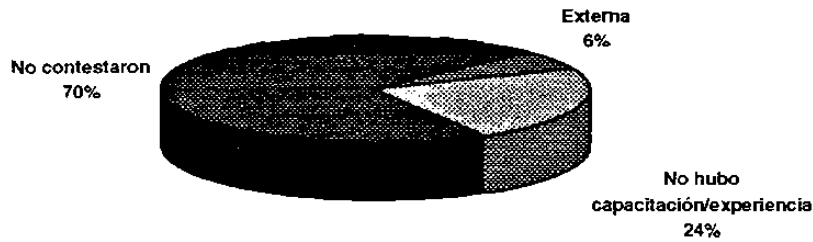
Fig.4.32 Empresas cuyas aplicaciones han sido desarrolladas por herramientas CASE



No se ha presentado una gran aceptación por parte de las empresas hacia las herramientas CASE, sea o no por la falta de cultura, por la típica resistencia al cambio o por la carencia de presupuesto, o bien por una combinación de todos estos factores: el área de sistemas debe de afrontar

los cambios en la tecnología que soporta el desarrollo de sistemas y más aún, si está comprobado que ayuda a mejorar la calidad y la productividad. Mas sin embargo no se olvide que la sola tecnología no constituye la panacea o la solución a nuestros problemas.

Fig.4.33 Tipo de capacitación que la empresa recibe



Lo ideal es que se lleve a cabo una capacitación por una persona con experiencia reconocida en el uso y manejo de la tecnología de información de desarrollo de sistemas(consultores), de no ser así se aprendería sobre la práctica como es el caso de muchas empresas, lo cual conllevaría a un uso deficiente de la tecnología dado que no se explotaría en toda su capacidad y que el tiempo que se requeriría para obtener un máximo provecho de ésta, sería mucho mayor y afectaría a los proyectos que se estén desarrollando.

Cabe mencionar que la sección III de este cuestionario no fue adecuadamente contestada por la mayoría de las empresas por la falta de metodologías y herramientas CASE . En cuanto a las respuestas obtenidas en la sección IV serán de gran ayuda para plantear las estrategias a seguir en el siguiente capítulo, cumpliendo así con el objetivo principal de este proyecto .

Con respecto a la experiencia de CASE en las empresas quisiera dar a conocer algunas ideas importantes:

- Entre las 17 empresas sólo dos cuentan con herramientas CASE siendo, éstas EasyCase y System Architect.
- Los que utilizan EasyCASE son para desarrollo de nuevos sistemas, mientras que sólo se utiliza para soportar 25% de desarrollo de los nuevos sistemas.
- Sólo tienen de seis meses a 1 año de experiencia en el uso de estas herramientas, sin embargo, gracias a su uso se ha logrado incrementar la productividad, la calidad y la comunicación con el usuario.
- Para poder utilizarlas consideran primeramente conocer las metodologías estructuradas.
- Logra con su uso mayor productividad, calidad y la mejora de comunicación con el usuario.
- El uso de las herramientas ha sido impulsado en ambos casos por la gerencia de sistemas.
- Estas herramientas soportan a las siguientes metodologías:
EasyCASE: LBMS.
Architect: GSN, YDM, WML.
- El fracaso que pudiera obtenerse por el uso de esta herramienta sería por la falta de capacitación y por el impacto al cambio.

Conclusiones relacionadas con el resultado de la encuesta

Podemos comprobar gracias a la información que se recopiló de los cuestionarios aplicado a empresas de diferente giro, que en la mayoría de ellas en sus áreas de desarrollo de sistemas se requiere tanto un cambio de cultura computacional como la generación de una cultura de calidad para lograr solucionar los siguientes problemas:

- A pesar de que algunos consideran importante el uso de metodologías, técnicas y herramientas otros no creen en ellas o muchas veces, si cuentan con ellas no las utilizan.
- El personal en el área lleva a cabo múltiples actividades(análisis, diseño, programación etc.), no hay una definición adecuada de funciones para cada elemento del área de desarrollo.
- Carecen en la mayoría de las empresas de herramientas que automatizan el desarrollo de sistemas como CASE, en muchas de ellas ni conocen su significado o las confunden con algunos lenguajes de programación.
- Siguen construyendo sistemas con lenguajes de tercera generación y metodologías de cuarta o quinta generación.
- En parte de las empresas el interés está en entregar los sistemas lo más pronto posible sin considerar la calidad.
- No siguen ninguna filosofía de calidad o si la utilizan no están adaptadas al área de sistemas.
- No se capacitan en las nuevas tecnologías de información, aprenden echando a perder.
- Los sistemas en su mayoría carecen de documentación provocando con esto problemas al momento de efectuarse los mantenimientos.
- El problema del mantenimiento persiste dedicando más recursos a esta actividad acumulándose así el trabajo relacionado con los nuevos desarrollos.

Si el departamento de sistemas quiere que se le considere como el soporte administrativo para los usuarios, éste deberá ofrecer productos de calidad, los cuales serán resultado de una excelente explotación o uso de metodologías y tecnologías que aseguren la calidad.

En seguida se procederá a presentar los conceptos relacionados con la calidad en los sistemas, dando a conocer los diferentes elementos y determinantes para lograr la calidad terminando con el modelo propuesto, la guía que soportará el desarrollo de sistemas y las estrategias a considerar para alcanzar nuestra meta : "un sistema con calidad".

5. CALIDAD EN EL DESARROLLO DE SISTEMAS.

5.1. SISTEMAS: UN DEPARTAMENTO DE PRODUCCION.

El departamento de sistemas a través de su infraestructura tanto material como humana, satisface las necesidades reales de información que el usuario necesita, utilizando los medios y organización adecuados para dar los resultados.

Su objetivo básico se puede dividir en dos grandes grupos: La planeación y diseño de nuevos sistemas y la operación de los actuales (producción de información) por lo cual se le puede considerar un departamento de producción, ya que con la ayuda de sus diferentes áreas (desarrollo, soporte técnico, operación etc.) produce información, la información de calidad que desea cada usuario. Sus funciones pueden equipararse a las que se realizan en un departamento de producción.

Si observamos el siguiente diagrama podremos apreciar que las áreas que componen a ambos departamentos son semejantes, siendo la única diferencia actualmente el área de control de calidad.

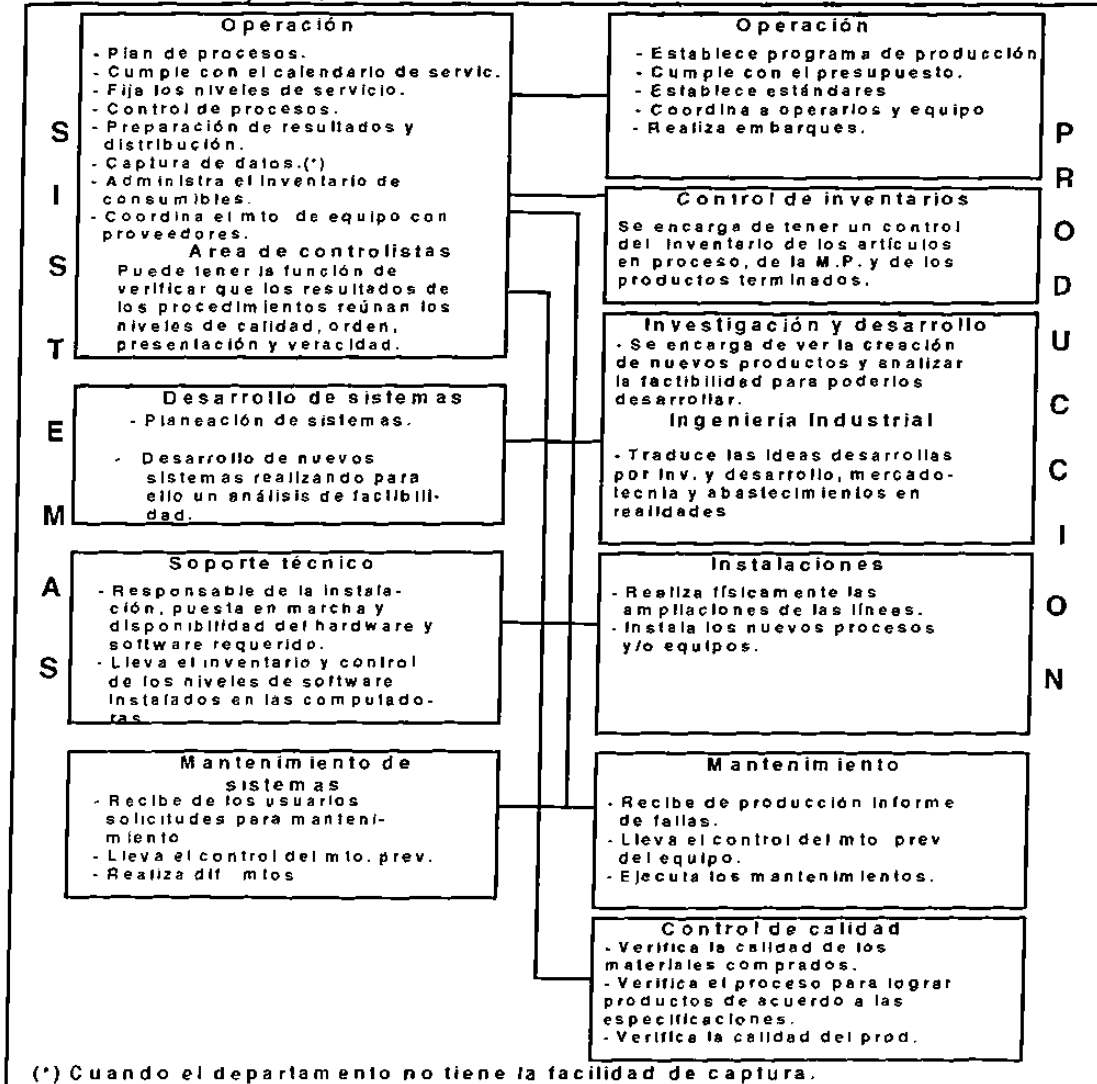


Fig. 5.1 Analogía entre un departamento de sistemas y producción

El control de calidad que en un departamento de producción se da es desde el momento que se recibe la materia prima hasta que se obtiene el producto terminado. Análogamente en el departamento de sistemas el control de calidad debería darse en la construcción de un sistema de información necesario para procesar los datos(materia prima) y obtener el producto terminado(información de calidad).

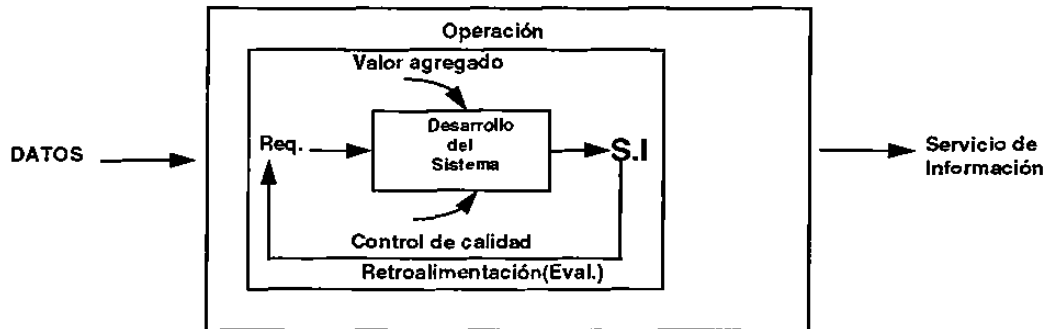


Fig. 5.2 El control de calidad como elemento indispensable en el desarrollo de un sistema de información

Esta calidad deberá irse generando desde la recepción de los requerimientos hasta generar el sistema, siguiendo un plan determinado con un elemento de control que detecte errores (diferencias entre lo programado y lo realmente obtenido), realizando cambios y ajustes necesarios para controlar así:

- Que la calidad del producto sea la adecuada.
- Que se produzca a la velocidad indicada.
- Que lo producido coincida con la demandado por los usuarios.
- Que los costos de producción sean adecuados.
- Que los requerimientos estén disponibles cuando se les necesite.

soportado todo el proceso por una excelente administración logrando así un producto adaptado a las necesidades y a los gustos del usuario.

Es importante mencionar todo esto, porque el control de calidad utilizado por las empresas productoras para el control de problemas en el proceso de producción, realmente no ha sido tomado en cuenta en la práctica del desarrollo de software; la realidad es que muchos creen saber el significado que la calidad representa confundiendo muchas veces con las pruebas, es decir muchos consideran que la calidad en el sistema se logrará corrigiendo los defectos en la programación, y entonces acaso no se toma en cuenta el grado de dificultad para utilizarlo, lo lento que es y la inflexibilidad que este presenta para realizar posibles cambios. Además, la falta de consideración de los distintos elementos que determinan y afectan la calidad seguirán ocasionando que los sistemas que se generan no cumplan con los requerimientos o criterios de calidad.

Es momento de pensar si ¿La calidad es un imposible para el área de desarrollo de sistemas?, ¿Realmente el personal de sistemas conoce el significado de ello, o carece de este tipo de cultura?

Para responder a estas interrogantes daré a conocer lo que para algunos autores representa la calidad en los sistemas y nuestro punto de vista acerca de ello.

Para responder a estas interrogantes daremos a conocer lo que para algunos autores representa la calidad en los sistemas, nuestro punto de vista acerca de ello y la realidad de ésta en algunas empresas de Monterrey.

5.2. CALIDAD EN EL SOFTWARE.

Calidad es:

Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo el software desarrollado profesionalmente. [PRE93].

El grado con el cual un producto de software posee un conjunto específico de atributos necesarios para cumplir con un propósito planteado [SC87].

El grado con el cual el usuario o cliente percibe que el software conoce sus expectativas y posee una combinación deseada de atributos.[SC87].

Totalidad de funciones o características de un producto o servicio relacionándose con su habilidad para satisfacer necesidades implícitas o declaradas[MOL,93].

Si analizamos todas estas definiciones y lo que representa la calidad para los diferentes autores mencionados en el capítulo dos, podríamos afirmar que la calidad es:

Un proceso que tiene un principio pero no un fin, implica cumplir con los requerimientos implícitos y explícitos del usuario, de tiempo, de costo, de funcionalidad y otros más, siendo lo más importante el valor que represente para la persona (cliente o usuario) el producto o servicio generado, soportado por estándares y tecnología de información que aseguren profesionalidad y permanencia.

Su prioridad para la época actual en que vivimos es alta ya que se considera indispensable para lograr la competitividad a nivel internacional, así como para establecer excelentes relaciones con los clientes o usuarios tanto internos como externos a la empresa. Es por eso que el área de desarrollo de sistemas de cualquier empresa debe darle la importancia que se merece tanto la calidad como a la

productividad dejando atrás aquel desarrollo para sólo cumplir en ese momento con los requerimientos del usuario sin tomar en cuenta diversos factores. Quizá forme parte del área de desarrollo de sistemas deberá tener presente :

- Alcanzar los objetivos del negocio.
- Lograr una mejora continua en la gente, los procesos y productos que forman parte del CVDS.
- Desarrollar en forma rápida(mejorando así el tiempo del CVDS).
- Anticipar y exceder los requerimientos del usuario.
- Reducir costos.
- Construir un software de fácil mantenimiento(flexible).
- Y un sistema tan amigable al usuario (fácil de aprender y utilizar).

Es decir su propósito desde ahora en adelante será lograr "*calidad en un sistema*".

La base de esto será la creación de una cultura en calidad así como una cultura computacional en nuevas tecnologías de información ya que de esta manera lograremos que los sistemas después de su liberación sufran pocas modificaciones disminuyendo así los costos y el tiempo comúnmente dedicado a la actividad del mantenimiento.

Cabe aclarar que al no cumplir su propósito el software o sistema contará con errores, sea este cualquier tipo de sistema (sistemas de tráfico aéreo, sistemas militares, automatización industrial, etc.) pudiendo causar no sólo daños sino quizás hasta la pérdida de vidas humanas. Es por eso que debemos de considerar que en todo proceso de desarrollo de software deben existir controles para evitar la propagación de errores y así lograr un sistema con calidad.

Sin embargo, hay que reconocer que en nuestra localidad en la mayoría de los departamentos de desarrollo de sistemas existen todavía muchas fallas en este proceso de desarrollo de sistemas, por la falta de capacitación en nuestro personal, esto ocasiona en la mayoría de los casos disgustos e insatisfacción del usuario, pérdida de oportunidades e incumplimientos en costos y tiempo. Por lo tanto a través de este capítulo y el siguiente, se irá dando a conocer según nuestra manera de pensar cuales son aquellos elementos y factores que debemos tomar en cuenta si hemos decidido desarrollar un sistema con calidad: siendo estos elementos y factores a considerar parte del proceso del desarrollo de un sistema y parte del resultado de ese proceso. Además, se mencionarán todas aquellos problemas y sus causas que podrían obstaculizar el proceso de desarrollo de un sistema con calidad, ya que a través de su conocimiento y llevando a cabo el curso de acción adecuado, podremos llegar a cumplir los siguientes objetivos de calidad:

OBJETIVOS DE CALIDAD

- * Identificar y satisfacer las necesidades de los usuarios.
- * Monitorear la calidad a través de revisiones en el proceso de desarrollo de sistemas.
- * Mejora de procesos a través de técnicas de prevención de errores.
- * Y lograr que los costos sean eficaces con el fin de eliminar los derroches.



Fig. 5.3 Los objetivos de calidad en el área de desarrollo de sistemas

Al perseguir estos objetivos y considerar los diferentes elementos como tecnología, asesorías, medio ambiente etc., podremos lograr dar valor a los procesos del negocio.

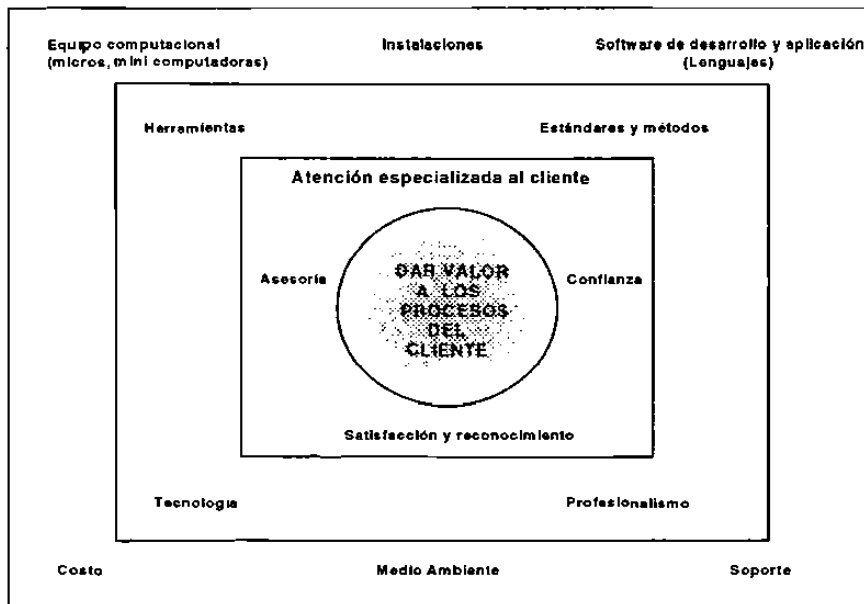


Fig. 5.4 Elementos indispensables para dar valor a los procesos del cliente

5.3. HACIA LA CALIDAD: SUS DETERMINANTES Y FACTORES.

La calidad de un sistema no estará determinada por la sola independencia de ciertos elementos (mercado, recursos humanos, recursos financieros, administración, insumos, tecnología, medio de trabajo) sino por la coordinación que se logre entre ellos, con el fin de que se den aquellos factores indispensables en un sistema con calidad.

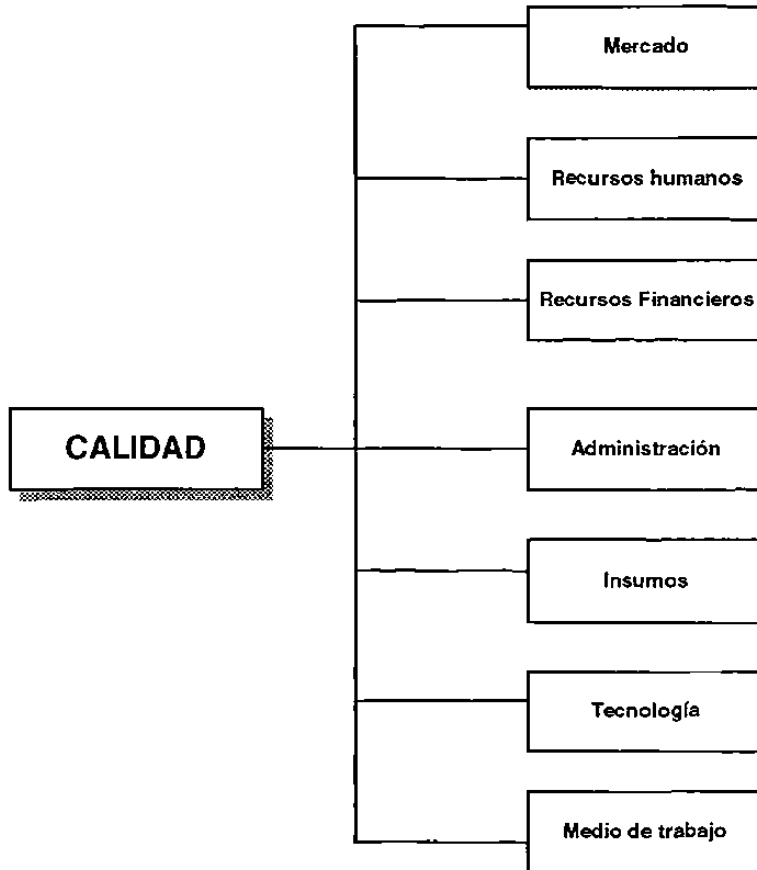


Fig. 5.5 Determinantes de la calidad

El mercado: Es importante considerar que el área de desarrollo de sistemas debe preocuparse por el medio ambiente competitivo que lo rodea y por la calidad de los productos que deberá producir, ya que un usuario podrá tomar la decisión en un momento dado de contratar los servicios de una consultoría externa.

Recursos humanos: Debido a los avances tecnológicos y a la búsqueda de una satisfacción total del cliente, es indispensable contar con personal lo suficientemente educado y capacitado; gente que tenga la facilidad de integrarse a un equipo, que pueda compartir y recibir ideas, además deberán ser personas capaces de mejorar su trabajo y aprender nuevas formas de ejecutarlo.

Recursos financieros: El departamento de sistemas deberá contar con los recursos necesarios para poder soportar el proceso de desarrollo de sistemas, llevando a cabo inversiones en tecnología de información, siempre y cuando su costo sea justificable, comparado con los beneficios que represente, siendo los principales beneficios a obtener: la calidad y productividad, así como la confiabilidad en el producto.

Administración: La calidad de un producto dependerá del compromiso y decisiones de la administración así como de la percepción que éstos tengan de la calidad, ya que sin el apoyo de la administración no se podrían realizar las actividades necesarias para desarrollar el software de calidad, por la inversión de recursos que se necesita.

Insumos: La calidad del producto estará ligada a ellos, siendo los requerimientos los insumos más importantes los cuales deberán de chequearse cuidadosamente, ya que el diseñador del sistema trabajará sobre ellos.

Tecnologías: Tanto el hardware, como las diferentes metodologías, técnicas y herramientas para el logro de la calidad y para el desarrollo del sistema, ayudarán a obtener esa calidad y productividad.

Medio de trabajo: Se requiere que el lugar en donde se lleve a cabo el desarrollo de un sistema sea ordenado, sin ruido excesivo y con excelente iluminación.

Es indispensable construir la calidad paso a paso siendo ésta soportada por un equipo de personas con una relación excelente entre ellos y con el usuario.

5.4. LOS FACTORES DE CALIDAD.

A continuación se presentan los factores de calidad del software de McCall enfocados principalmente a las características operativas, la capacidad de soportar cambios y a la adaptabilidad a nuevos entornos[PRE93] siendo éstos:

Corrección: Grado en que un programa satisface las especificaciones y los objetivos del usuario.

Confiabilidad: Grado en que un programa lleve a cabo sus funciones esperadas con la precisión requerida.

Eficiencia: Cantidad de recursos computacionales y de código requerido por un programa para llevar a cabo sus funciones.

Integridad: Grado en que puede controlarse el acceso al software o a los datos por personas no autorizadas.

Facilidad de uso: Esfuerzo para aprender, utilizarlo, preparar entradas e interpretar las salidas de un programa.

Facilidad de mantenimiento: Esfuerzo requerido para localizar y corregir un error en un programa.

Flexibilidad: Esfuerzo requerido para modificar un programa operativo.

Facilidad de prueba: Esfuerzo requerido para probar que un programa ejecuta su función comprometida.

Portabilidad: Esfuerzo requerido para transferir un programa desde un hardware y/o un entorno de sistemas de software a otro.

Reusabilidad: El grado en que un programa puede ser utilizado en otras aplicaciones.

Facilidad de interoperación: Esfuerzo requerido para unir un sistema con otro.

Otra manera de clasificar estos factores de calidad son los que propone Freeman, el cual los divide en dos grupos: el primero de los grupos es el que caracteriza a un producto con una calidad básica y el segundo grupo el que hace referencia a productos con una calidad extra [FRE,87].

CALIDAD BASICA	CALIDAD EXTRA
FUNCIONALIDAD	FLEXIBILIDAD
CONFIABILIDAD	REPARABILIDAD
FACILIDAD DE USO	ADAPTABILIDAD
ECONOMIA	DOCUMENTACION
SEGURIDAD	REALZABILIDAD
	COMPRESIBILIDAD

Tabla 5.1 Los factores de la calidad

Funcionalidad: El software puede ejecutar las funciones que fueron requeridas por la tarea para la cual éstas se destinaron.

Confiabilidad: El sistema debe ser capaz de ejecutar sus funciones consistentemente tal como se destinaron.

Facilidad de uso: El sistema debe de permitir a los usuarios a trabajar en conceptos y términos familiares para ellos, aprenderlo a usar con un mínimo de entrenamiento y deberá proporcionar un fácil acceso a la información que se requiera.

Economía: El software deberá realizar su trabajo con un mínimo de recursos.

Seguridad: La operación del software no debe permitir dañar o destruir sobre lo que tiene control.

Flexibilidad: Lo es si éste puede manejar un rango de diferentes entradas.

Reparabilidad: La habilidad de un sistema para corregir errores.

Adaptabilidad: La habilidad de un sistema para operar bajo un rango de condiciones.

Documentación: La extensión de información acerca del sistema y su uso es capturado y presentado claramente.

Realzabilidad: La facilidad de añadir nuevas capacidades a una pieza existente de software.

Comprensión: Es entendible el sistema si se comprende que hace y como usarlo con un mínimo de esfuerzo.

Si analizamos todos estos factores que presentan ambos autores podríamos establecer una relación entre ellos por su semejanza:

McCall	Freeman
Corrección	Funcionalidad
Confiabilidad	Confiabilidad
Eficiencia	Economía
Integridad	Seguridad
Facilidad de uso	Facilidad de uso
Facilidad de mantenimiento	Reparabilidad
Flexibilidad	Flexibilidad
Portabilidad	Adaptabilidad

Tabla 5.2 Relación entre los factores de McCall y Freeman

La presencia de algunos de estos factores de calidad tanto en el producto, proceso y en la documentación generada durante el desarrollo del sistemas, estará determinada a través de métricas o medidores, siendo éstos los que determinarán si se ha logrado construir un sistema con calidad.

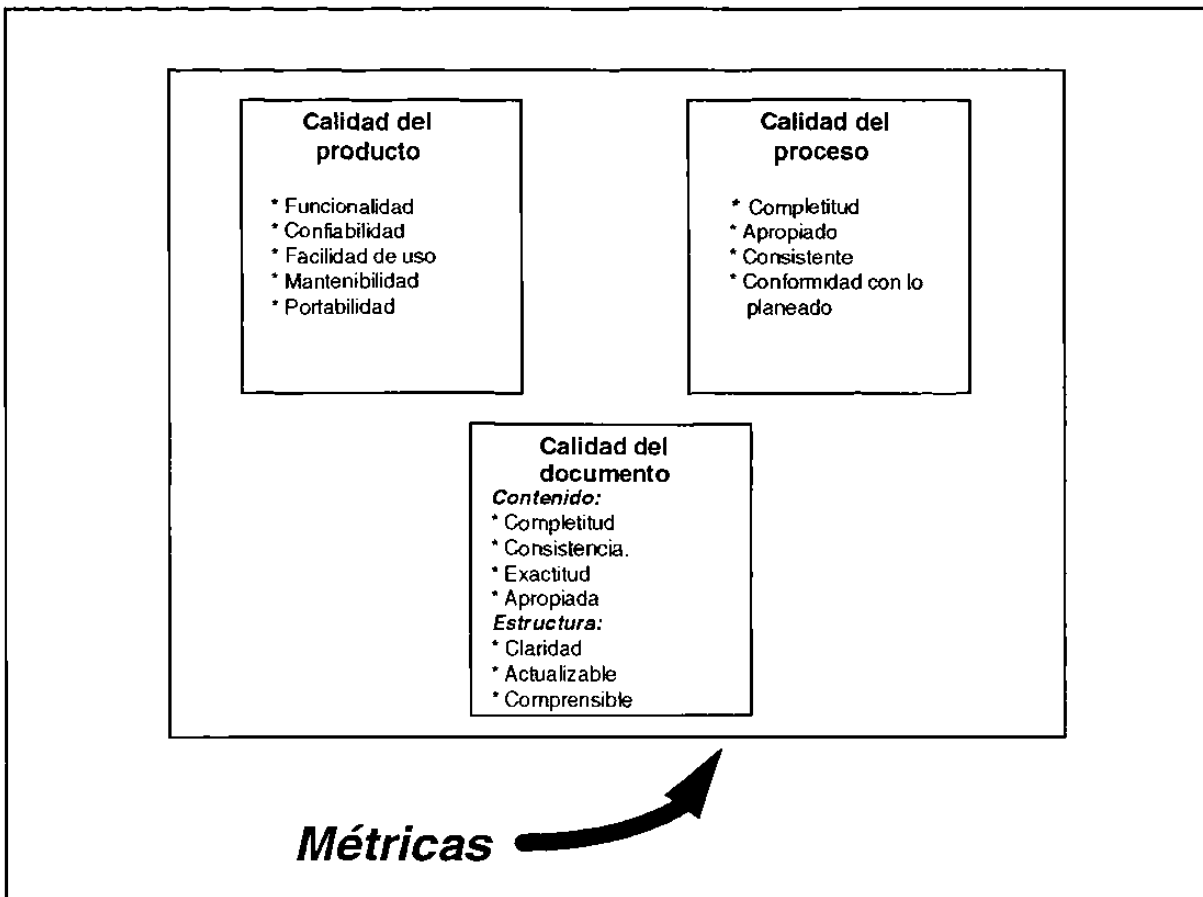


Fig. 5.6 Las métricas como medidores de la presencia de los factores de calidad

5.5. METRICAS.

Las métricas se consideran como técnicas para mejorar tanto la calidad como la productividad del software, y son aplicables a los factores de calidad que se buscan obtener tanto en el proceso como en el producto final del desarrollo del sistema. Ellas lograrán a través de mediciones mantener el proyecto de desarrollo del sistema dentro del plan, y soportarán además los objetivos de la calidad.

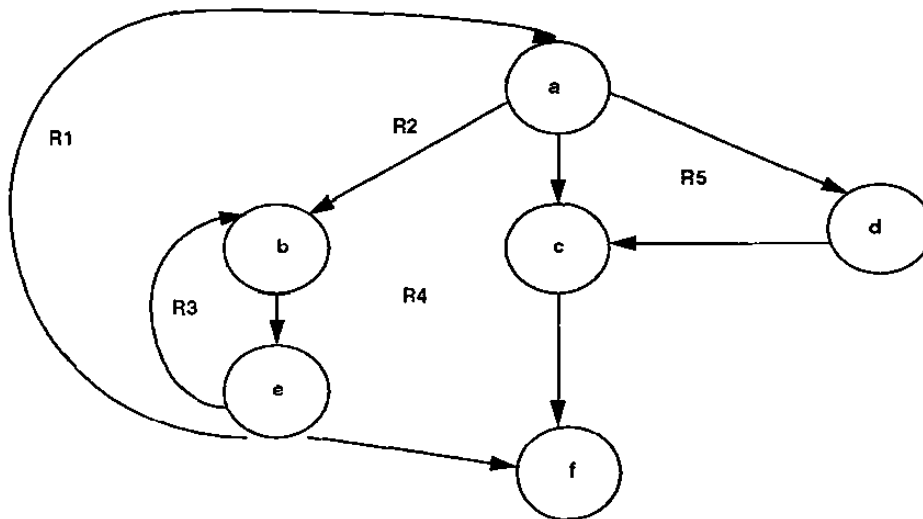
Existen métricas limitadas a ciertos factores como métricas que se enfocan a la mayoría de éstos tal es el caso de las métricas de McCall.

A continuación se explicarán algunas de ellas considerando los puntos de vista de sus autores:

McCabe y su métrica.

Para medir la complejidad del software es necesario representar esa complejidad del software realizando un diagrama del programa. Este diagrama mostrará las tareas de procesamiento y el flujo de control. Así la medida de la complejidad estará basada en la complejidad ciclomática del diagrama. Para ello será necesario establecer la cantidad de regiones las cuales estarán determinadas por las áreas limitadas y no limitadas que rodean al diagrama. Este número de regiones ayudará a determinar la dificultad para realizar pruebas, siendo esta medida relacionada también con la confiabilidad.

El recomienda que si el número de regiones es de 10 será un límite para el tamaño de cada uno de los módulos del programa, de no cumplir con este límite la etapa de pruebas será difícil de realizar [PRE93].



Cantidad de regiones=5.
Tareas=a,b,c,d,e,f

Fig. 5.7 Diagrama utilizado para medir la complejidad de un programa

McCall define un conjunto de métricas utilizadas para desarrollar expresiones para cada uno de sus factores de acuerdo a la siguiente relación:

$$F_c = c_1 \times m_1 + c_2 \times m_2 + \dots + c_n \times m_n$$

donde F_c representa un factor de calidad del software y c_n son los coeficientes de regresión y m_n son las métricas que afectan al factor de calidad. Muchas de las métricas definidas por McCall solo pueden medirse en forma subjetiva por eso él propone un esquema de graduación que va de una escala de 0(bajo) al 10(alto) y utiliza las siguientes métricas para este esquema [PRE,93]:

Facilidad de auditoría: La facilidad que se puede corroborar la conformidad con los estándares.

Exactitud: La precisión de los cálculos y el control.

Normalización de las comunicaciones: El grado en que se usan el ancho de banda, los protocolos y las interfases estándar.

Completitud: El grado en que se ha conseguido la total implementación de las funciones requeridas.

Concisión: Lo compacto que es el programa en términos de líneas de código.

Consistencia: El uso de un diseño uniforme y de técnicas de documentación a lo largo del proyecto de desarrollo de software.

Estandarización de los datos: El uso de estructuras de datos y de tipos estándar a lo largo de todo el programa.

Tolerancia de errores: El daño que se produce cuando el programa encuentra un error.

Eficiencia en la ejecución: El rendimiento en tiempo de ejecución de un programa.

Facilidad de expansión: El grado en que se puede ampliar el diseño arquitectónico, de datos o procedimental.

Generalidad: La amplitud de la aplicación de potencial de los componentes del programa.

Independencia del hardware: El grado en el que el software es independiente del hardware sobre el que opera.

Instrumentación: El grado en el que el programa muestra su propio funcionamiento e identifica errores que aparecen.

Modularidad: La independencia funcional de los componentes del programa.

Facilidad de operación: La facilidad de operación de un programa.

Seguridad: La disponibilidad de mecanismos que controlen o protejan los programas o los datos.

Autodocumentación: El grado en que el código fuente proporciona documentación significativa.

Simplicidad: El grado en que un programa puede ser entendido con dificultad.

Independencia del sistema de software: El grado en que el programa es independiente de las características del sistema operativo y de otras restricciones del entorno.

Facilidad de traza: La posibilidad de seguir la pista de la representación del diseño o de los componentes reales del programa hacia atrás, hacia los requisitos.

Formación: El grado en que el software ayuda para permitir que nuevos usuarios apliquen el sistema.

Su relación con los factores antes definidos estaría determinada de la siguiente forma:

Métricas	Factor de calidad
Exactitud y tolerancia de errores	Confiabilidad
Compleitud, Concisión, Eficiencia en la ejecución	Eficiencia
Consistencia	Mantenibilidad
Estandarización en los datos, Modularidad, Generalidad	Reusabilidad
Facilidad de expansión, formación	Flexibilidad
Independencia de hardware	Interoperabilidad
Facilidad de operación, simplicidad	Facilidad de uso
Seguridad	Seguridad
Facilidad de traza	Prueba
Independencia del sistema	Portabilidad

Tabla 5.3 Relación entre las métricas y los factores de calidad

Tomando como base las métricas anteriores vemos que las métricas de McCall se enfocan a todos los factores de la calidad de software (Eficiencia, confiabilidad, Flexibilidad, Realzabilidad, Facilidad de uso, Seguridad, Interoperabilidad, Portabilidad, Mantenibilidad, Reusabilidad, Prueba) excepto al de funcionalidad mientras que McCabe sólo se enfoca al factor de confiabilidad y pruebas.

El fin de utilizar estas métricas en el proceso del desarrollo de un sistema permitirá:

- * Mejorar tanto la calidad del proceso como del producto.
- * Mejorar la satisfacción del cliente.
- * Y mejorar la cultura de calidad.

5.6. ESTANDARES.

Los estándares son las reglas o bases para llevar a cabo una comparación, utilizada para asegurar el tamaño, contenido, valor o calidad de un objeto o actividad, muchas veces son definidos por la metodología de desarrollo en otras ocasiones por el departamento de sistemas, o se siguen estándares definidos por algún instituto como la IEEE.

Gracias a ellos, es posible reducir el número de defectos y los costos durante el desarrollo de un sistema además de ayudar a que el mantenimiento a un sistema sea fácil de realizar.

Son muchos los estándares que se han definido como los relacionados con el diseño, programación, pruebas, documentación e inclusive algunos de ellos hacen énfasis a la calidad para el desarrollo de sistemas. Para mostrar algún ejemplo de ello se presentarán los estándares definidos tanto por la FAA, IEEE, e ISO los cuales tienen como objetivo cumplir con ciertos factores de calidad.

NOMBRE DEL ESTANDAR DOD(Departement of defense)	ENFOQUE: Factor de funcionalidad
1. Especificación de requerimientos de un programa de computadora y requerimientos de un programa de computadora de calidad para administrar la calidad del software. 2. Monitorear y evaluar la implementación de un programa de calidad del contratista incluyendo su efectividad. 3. Aceptación del gobierno sobre el software.	

Tabla 5.4 Estándar DOD con enfoque al factor funcionalidad

NOMBRE DEL ESTANDAR FAA(Federal Aviation Administration)	ENFOQUE: Aseguramiento de calidad
<p>Los puntos principales a considerar en este estándar son:</p> <ol style="list-style-type: none"> 1. Flujo del proceso del desarrollo. 2. Estándares de calidad de software. 3. Indicadores de estatus del proceso. 4. Registros de las actividades de calidad. <p style="text-align: center;">Resumen de FAA-STD-018</p> <ul style="list-style-type: none"> Organización. Flujo del proceso del desarrollo. Estándares de calidad en el software. Instrucciones del trabajo del software de computadora. Indicación de estatus del proceso. Administración de la configuración. Revisión del diseño. Acciones correctivas. Controles de prueba. Herramientas, Técnicas y metodologías Control del subcontratista Registros. Auditorías. 	

Tabla 5.5 Estándar FAA con un enfoque al aseguramiento de calidad

NOMBRE DEL ESTANDAR

IEEE

ENFOQUE:

Factor de funcionabilidad y confiabilidad, facilidad de uso, interoperabilidad y precio.

El propósito del estándar es proporcionar uniformidad, requerimientos mínimos de aceptación para preparación y contenido de los planes de aseguramiento de calidad. Este estándar se aplica a software crítico donde las fallas pueden impactar en la seguridad o causar grandes pérdidas financieras o sociales.

Resumen del estándar IEEE STD 730-1984

Plan para asegurar la calidad del software.

Propósito.

Documentos de referencia.

Administración.

Organización.

Tareas.

Responsabilidades.

Documentación.

Propósito.

Mínimos requerimientos de documentación.

Especificaciones de requerimientos de software.

Descripción de diseño de software.

Plan de verificación y validación del software.

Reporte de verificación y validación del software.

Documentación del usuario

Otros.

Estándares, Prácticas, Convenciones.

Propósito.

Contenido.

Revisión y auditorías.

Propósito.

Requerimientos mínimos.

Revisar los requerimientos del software.

Revisar el diseño preliminar.

Revisar el diseño crítico.

NOMBRE DEL ESTANDAR IEEE	ENFOQUE: Factor de <i>funcionabilidad</i> y <i>confiabilidad</i> , <i>facilidad de uso</i> , <i>interoperabilidad</i> y <i>precio</i> .
<p>continuación,....</p> <ul style="list-style-type: none">Revisar la verificación y validación del software.Auditorías Funcionales.Auditorías físicas.Auditorías en proceso.Revisiones empresariales.Administración de la configuración del software.Reportes de problemas y acciones correctivas.Herramientas, Técnicas y Metodologías.Control de código.Control del proveedor.Colección de registros, mantenimiento y retención.	

Tabla 5.6 Estándar IEEE con un enfoque a diversos factores de calidad

<p>NOMBRE DEL ESTANDAR</p> <p>ISO-9004</p>	<p>ENFOQUE</p> <p>CALIDAD EN EL PROCESO Y PRODUCTO.</p>
<p>La organización de estándares internacionales (ISO) a través de sus estándares 9000-9004 permiten el aseguramiento de la calidad en cualquier producto. La serie ISO 9000 esta formado por cinco documentos:</p> <p>ISO-9000: Administración de la calidad y estándares que aseguran la calidad : Guías para selección y uso. ISO-9001: Sistemas de calidad- Modelo para asegurar la calidad en el diseño/diseño/desarrollo, producción, instalación y servicio). ISO-9002: Sistemas calidad: Modelo para asegurar la calidad en la producción e instalación. ISO-9003: Sistemas de calidad - Modelo para asegurar la calidad en la inspección final y en las pruebas. ISO-9004: Administración de calidad y guías del sistema de calidad .</p> <p>Considerando a ISO-9004 como el estándar con el enfoque hacia la satisfacción del cliente, así como hacia la evaluación de riesgos y beneficios. Es el estándar adecuado para desarrollar e implementar un sistema de calidad ya que su calidad está orientada a tanto el proceso como el producto, quien a través de sus guías muestra los diversos factores(técnicos, administrativos y humanos) que pueden afectar la calidad del producto. Algunos de los párrafos claves en este estándar son:</p> <ul style="list-style-type: none"> * Verificación de las especificaciones o requerimientos de los productos terminados en cada etapa. Si esta verificación no es económicamente factible, verificar entonces sólo el producto final. * El método utilizado para asegurar la calidad en los requerimientos necesarios para la producción de un sistema de penderá de su importancia para la calidad. * Debe existir suficiente control en todos los sistemas de medición utilizados en el desarrollo e instalación para que exista una confianza en las decisiones sobre los datos de las mediciones. 	

Tabla 5.7 Estándar ISO-9004 con un enfoque a la calidad en el proceso y producto

5.7. INSPECCION O PREVENCION.

La inspección como parte del control de calidad tiene como objetivo verificar generalmente en la etapa final el cumplimiento de estándares o especificaciones deseadas en el sistema, pocas veces esta inspección se lleva a cabo en cada etapa de ciclo de vida del desarrollo de sistemas, logrando una detección tardía de fallas, obstaculizando el logro de la calidad.

Pero si se utilizara en cada etapa del CVDS, reconoceríamos que no se pueden obtener los productos de acuerdo a las especificaciones en cada etapa, de manera que habría retrabajos que ocasionarían costos y quizás por el hecho de que a nadie le gusta repetir un trabajo, los productos generados en cada etapa irían incrementando los defectos en el sistema. Todo esto conlleva a determinar que la inspección ni mejorará, ni garantizará la calidad.

Lo que debemos hacer es pasar de una inspección a una prevención, y asegurar así la calidad del sistema. Esta prevención deberá ser prospectiva, para que permita que las cosas se hagan bien a la primera, a través de una identificación de errores y sus causas, logrando que éstos no ocurran otra vez, llevando a cabo ciertas acciones para prevenirlos.

Esta prevención enfocada al proceso de desarrollo del sistema permitirá producir desde un inicio una calidad que sea clara y aceptable por el usuario para que no existan rechazos posteriores.

Asimismo, la prevención contribuirá a que los errores no ocurran de nuevo mejorando el proceso y disminuyendo los costos y retrabajos.

Su implementación en el desarrollo de sistemas implicará :

- Reportar defectos.
- Analizar las causas.
- Desarrollar estrategias.
- Implementarlas.
- Determinar cuál es su desempeño.
- Retroalimentación.

Tanto el uso de estándares, como de algún *checklist*(guía de actividades, productos por cada etapa) podrán ser útiles en la prevención de errores.

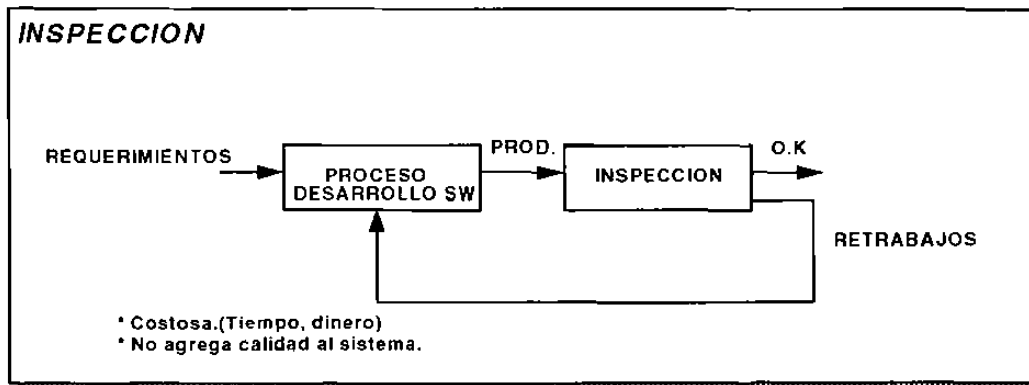


Fig. 5.8 La calidad por inspección

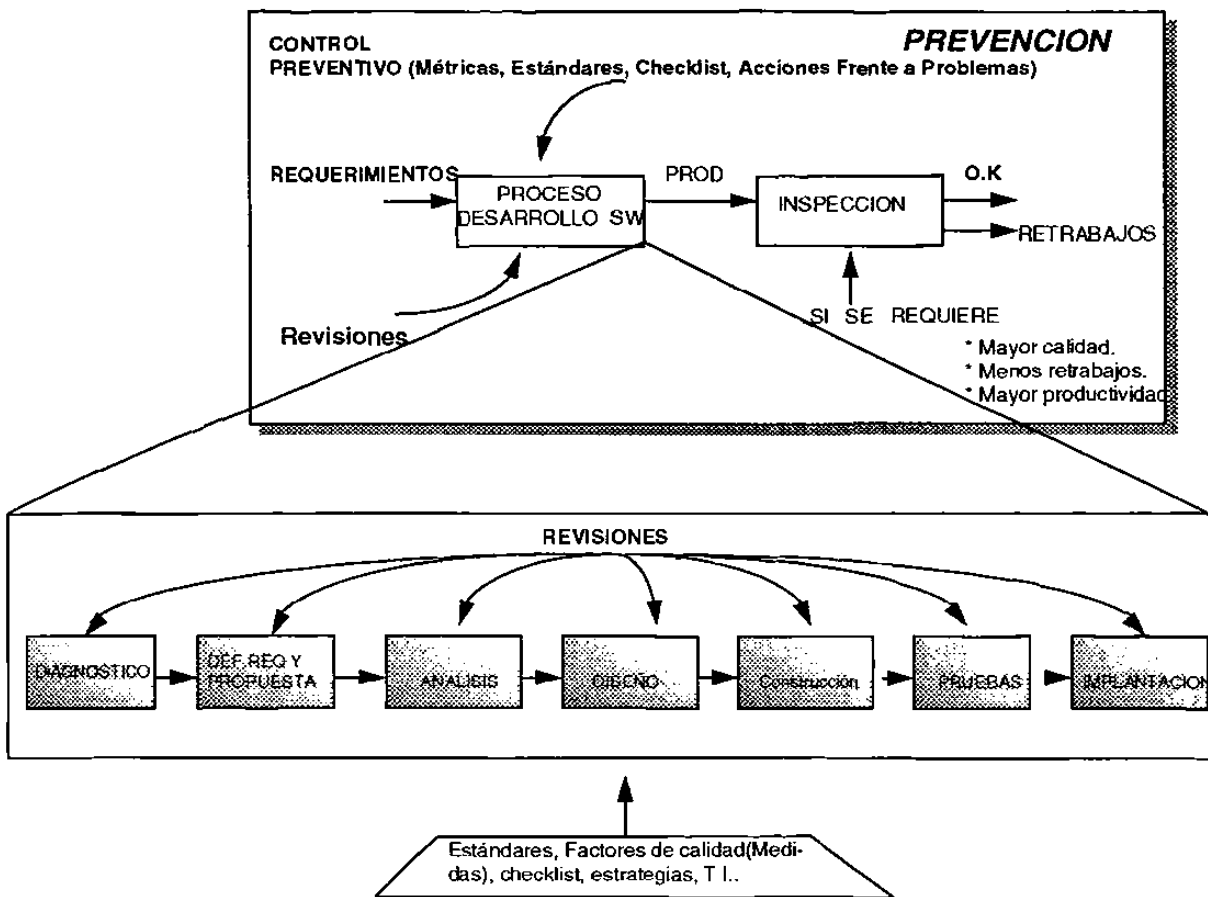


Fig. 5.9 La calidad por prevención básica en el desarrollo e implantación de un sistema.

5.8. EL FACTOR HUMANO.

La calidad que puede llegarse a obtener en el producto dependerá no sólo del trabajo de cada persona, o de la tecnología que éstas pudieran utilizar, sino del trabajo que se logre en equipo, cuyas personas además de estar capacitados y educados en metodologías, técnicas y herramientas (cultura computacional) deberán comprender y sentir la calidad (cultura de calidad), a ser lograda a través de una confianza, disciplina y compromiso. Dicho compromiso implicará una mejora continua y con él cual se logrará el intercambio de información y experiencias, logrando que las personas tengan la oportunidad de saber quién hace algo mejor para despertar el deseo de saber cómo hacerlo.

La calidad no se dará por sí misma debe hacerse que suceda a través de los recursos humanos quienes contando con los medios necesarios y además conociendo no sólo lo que están haciendo, sino la manera como pueden prevenir los errores para mejorar su trabajo, permitirán alcanzar los objetivos de calidad tanto del departamento de sistemas como de la misma empresa.

Los administradores deberán apoyar a los recursos humanos, proporcionándoles una capacitación adecuada para posteriormente elegir aquéllos que cuenten con las siguientes características necesarias para llevar a cabo un trabajo con calidad en el área de desarrollo de sistemas:

- Habilidad para ejecutar el trabajo.
- Interés en el trabajo.
- Entrenamiento recibido.
- Experiencia en metodologías, técnicas y herramientas.
- Habilidad para comunicarse con otros.
- Habilidad para compartir responsabilidad con otros.
- Habilidades administrativas.
- Capacidad de análisis.
- Conocimiento del negocio.
- Capacidad de entrevistar.
- Saber escuchar.
- Ser proactivo.
- Ser una persona con calidad de vida.
- Trabajar bajo presión.
- Humilde.

Las funciones y habilidades del personal involucrado en el desarrollo del sistema con calidad.

Usuario o cliente: A través de una solicitud al departamento de sistemas el usuario define el problema a solucionar o el área de oportunidad a explotar, fijando los objetivos del proyecto y los beneficios a lograr. El es quién podrá decidir por su propia autoridad o con autorización de sus superiores, que el proyecto se inicie después de que el encargado del proyecto del área de desarrollo de sistemas le presente la propuesta, durante la ejecución del proyecto recibirá información acerca de el grado de avance del proyecto y del cumplimiento de los objetivos fijados previamente decidiendo si se requiere una negociación de los mismos o parar el proyecto.

Al terminar el proyecto el usuario lo aprobará si cumple con los objetivos previamente establecidos.



Habilidades

- Habilidad de comunicación verbal y escrita.
- Tomador de decisiones.
- Conocimiento del negocio.
- Capacidad de reconocer problemas.
- Conocimiento de sus necesidades y futuros requerimientos.

Administrador: Encargado de determinar las actividades a desarrollarse en el proyecto, definiendo para cada una de ellas los responsables y las fechas de inicio y terminación, además realizará un monitoreo y control para reportar el progreso del proyecto y el cumplimiento de los objetivos.



Habilidades

- Conocimiento de la empresa.
- Dominio de la tecnología asociada al proyecto.
- Habilidades para planear, monitorear y controlar el proyecto.
- Tomador de decisiones.
- Capacidad de trato y de relación.
- Capacidad de negociación.

Analista: Participa en el desarrollo de nuevos proyectos y en la adecuación a proyectos ya existentes, determinando a través de una análisis de la necesidad del usuario y del sistema actual si su problema puede ser resuelto por medio de un desarrollo de sistemas, planteando así la propuesta de desarrollo encaminada a los requerimientos del cliente.



Habilidades

- Creatividad para dar nuevas soluciones a problemas antiguos y adaptar las soluciones antiguas a problemas existentes.
- Analizar los problemas y tomar decisiones.
- Organizado.
- Habilidad de comunicación oral y escrita.
- Conocimiento de metodologías, técnicas y herramientas que soportan el análisis (Modelación).
- Proactivo.
- Futurista.

Diseñador: Representa el sistema a través de un modelo a ser construido posteriormente.



Habilidades

- Habilidad de comunicación.
- Conocimiento de técnicas y herramientas de análisis y diseño.
- Habilidad para interpretar el análisis de sistemas.
- Ordenado.
- Creativo.

Programador: Participan en diferentes proyectos siendo su principal objetivo realizar la programación de sistemas.



Habilidades

- Saber interpretar el análisis y diseño de un sistema.
- Conocimiento en metodologías, técnicas y herramientas de programación.
- Facilidad de comunicación.

Audidores: Son quienes a través de una revisión sistemática y formal verifican el cumplimiento de estándares y controles establecidos para el desarrollo de un sistema, su documentación y su administración, generando como resultado de la revisión, un reporte claro y completo de las deficiencias que se detectaron.



Habilidades

- Conocimiento de la organización.
- Conocimiento de auditoría y de tecnología de información.
- Trabajar con la gente.
- Instinto de auditor.

Es recomendable que cuando se cuenta con herramientas CASE que se dominan y que se utilizan para automatizar todo el ciclo de desarrollo de un sistema no involucre a tanto personal en el proyecto. Sin embargo, dado que este tipo de herramientas está comenzando a penetrar en la industrias de México, tendrán que seguirse desempeñando las funciones que en la mayoría de las empresas de México existen, claro está que esto tendrá que cambiar conforme se vaya adoptando la nueva tecnología.

<i>Si cuenta con una herramienta CASE</i>	<i>Si utiliza cualquier otro tipo de herramientas</i>
<ul style="list-style-type: none">- Administrador.- Analistas capacitados en interfaces y modelación de datos y procesos.- Auditor.	<ul style="list-style-type: none">- Administrador.- Analista.- Diseñador.- Programador.- Auditor.

Tabla 5.8 Personal necesario en el desarrollo de un sistema utilizando herramientas CASE u otro tipo de herramientas

5.9. MODELO PROPUESTO.

A través de el siguiente diagrama podemos observar en forma resumida todos aquellos elementos que consideramos que están relacionados con cada uno de los determinantes de la calidad y que son indispensables en el proceso de desarrollo de un sistema. Estos elementos así como la nueva forma de administrar siendo una filosofía de calidad en la cual se han incluido los puntos más importantes del programa de mejoramiento de Crosby y las catorce obligaciones de la alta administración propuestas por Deming.

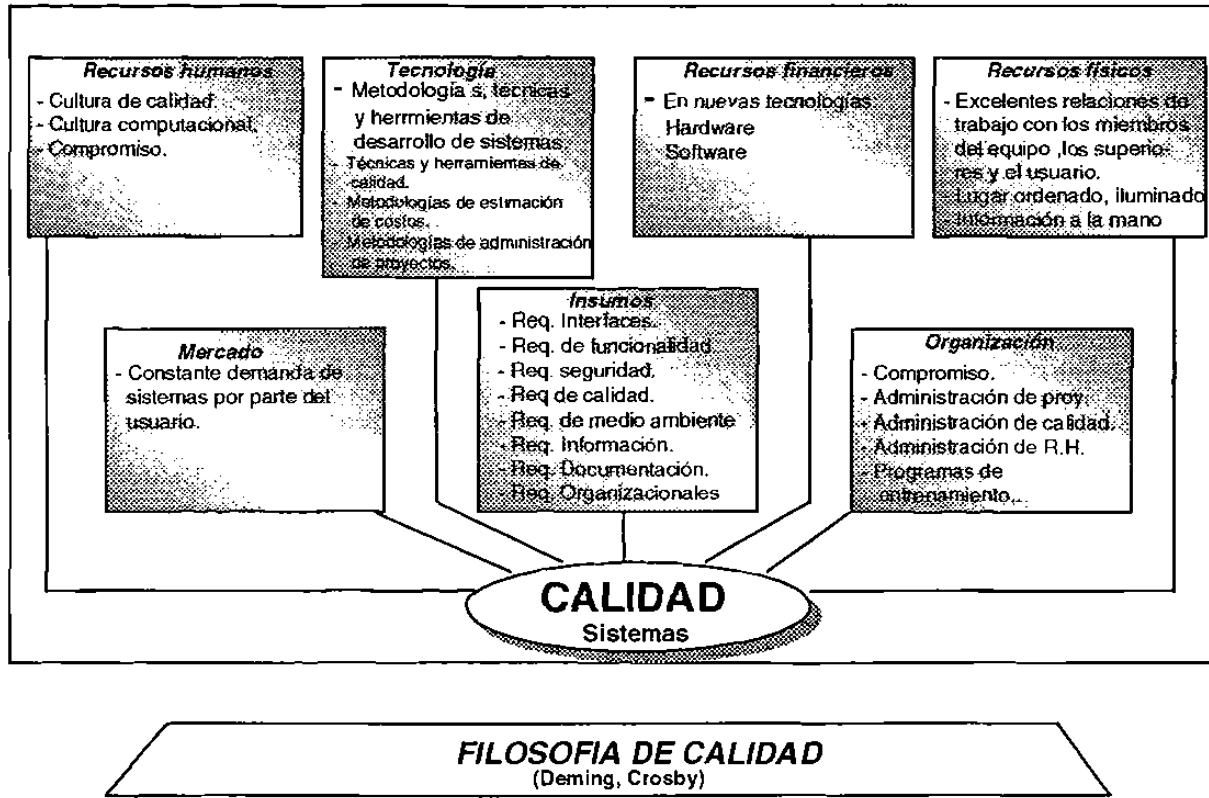


Fig. 5.10 Determinantes y sus componentes para el logro de un sistema con calidad

Los determinantes contribuirán a mejorar el proceso de desarrollo del sistema, produciendo un sistema de calidad a menor costo. Una vez lograda la calidad así como niveles aceptables de productividad, se proseguirá a conseguir altos los niveles de calidad que conlleven a una alta productividad, así como una mayor reducción de costos en todo el proceso, siendo esto resultado de un mejoramiento continuo de la calidad.

El resultado de unir las dos filosofías adaptándolas al área de sistemas es el siguiente:

- a) Crear constancia de propósito dirigida a mejorar los productos y los servicios.
 Crear constancia en el propósito de mejorar el proceso de desarrollo de sistemas con el fin de lograr un producto de calidad. De tal suerte que a través de un plan a seguir para cumplir los objetivos de calidad se realice:

- Una asignación de recursos.
 - Innovaciones.
 - Mejoras en el proceso de desarrollo de sistemas.
 - Capacitación y entrenamiento.
- b) Adoptar la nueva filosofía.
Será necesario un cambio de cultura tal que todas aquellas personas involucradas en el proceso de desarrollo de un sistema puedan hacer bien su trabajo haciendo uso de apropiadas metodologías, técnicas y herramientas. La nueva cultura no deberá aceptar errores en el desarrollo de sistemas, demoras así como tanto los típicos rechazos de usuario y mantenimientos interminables.
- c) Compromiso de la administración con la mejora de calidad.
La administración debe demostrar su compromiso y participación en el proceso y en la nueva filosofía de calidad. Tendrá la responsabilidad de educar, entrenar y vender la idea de la calidad continuamente. Deberá tener paciencia en su compromiso hacia la calidad ya que el cambio cultural requerirá de un proceso a largo plazo. Además guiará a los desarrolladores hacia el enfoque de servir al cliente ayudándolos a mejorar el proceso de desarrollo del sistema, a través de una retroalimentación hacia la solución de problemas encontrados.
- d) Educación y entrenamiento.
Las personas involucradas en el desarrollo de sistemas deberán estar convencidos y comprometidos con la nueva filosofía, para que con el paso del tiempo esta sea considerada como una nueva forma de trabajo. Asimismo, es necesario un entrenamiento y capacitación continua para que utilicen correctamente la tecnología que tienen a su alcance, y así se mejore la calidad y la productividad.
- e) Formación de equipos.
Es necesario crear un equipo de personas para lograr la calidad y mejorarla a través del proceso de desarrollo de sistemas. Las personas que formen parte del equipo que asegure la calidad (si la empresa puede soportar los costos, se puede tener un auditor y un experto en ingeniería de software) como las del equipo de desarrollo de sistemas, deberán aprender a trabajar en equipo logrando a través de los esfuerzos individuales una mejora en el proceso. Definirán entre todos las acciones necesarias para mejorar el programa de calidad.
- f) Instituir el liderazgo.
Se requiere que el líder de proyectos ayude a los desarrolladores a realizar su trabajo en forma correcta asegurándose de que cuenten con las metodologías, técnicas y herramientas necesarias.
- g) Fijar objetivos.
El establecimiento de objetivos es básico para todo proyecto de desarrollo de sistemas, ya que a través de ellos el líder de proyectos podrá tener un control sobre el proyecto y tomar decisiones eficaces y coherentes. Estos objetivos a su vez estarán ligados con acciones a realizar por los miembros del equipo de desarrollo, y con las medidas que permitirán evaluar el avance o el cumplimiento de esos objetivos.

h) Día de cero defectos.

Debemos empezar estableciendo un día de cero defectos, motivando a aquellas personas que cumplan en un 100% con los objetivos de calidad y del proyecto, quizá logremos que todos los días sean de cero defectos.

i) Eliminar "slogans".

No se requieren exhortaciones que generen frustraciones por no poder lograrlas. Es mejor proporcionar guías adecuadas para lograr la calidad en el sistema. Además las metas que se definan deberán ir acompañadas de sus metodologías para alcanzarlas.

j) Terminar la dependencia de la inspección masiva para lograr la calidad.

Como mencionamos anteriormente inspeccionar siempre será tardío, poco confiable, costoso e inefectivo. La calidad no se logrará por la inspección sino por el mejoramiento que se logre en el proceso del desarrollo del sistema (prevención). Es indispensable anticipar problemas conociéndolos al igual que sus causas. Recuerde que la inspección final de lo que ya hizo y lo que no pudo ser resulta ser muy tardía.

k) El costo de la calidad.

Todos los involucrados en el proceso de desarrollo conocen el costo de la no calidad, el cual estará asociado a no hacer las cosas bien a la primera vez y por el no cumplimiento de los objetivos tanto del proyecto como de calidad. Al realizar las cosas bien desde un principio no tendrá que incurrir en gastos extras para arreglar o hacer de nuevo los productos generados en cada etapa del desarrollo de sistemas. Ahora bien, se deberán cuantificar los costos de la calidad para determinar si ésta será factible económicamente.

l) Acabar con la práctica de otorgar negocios en base a precios.

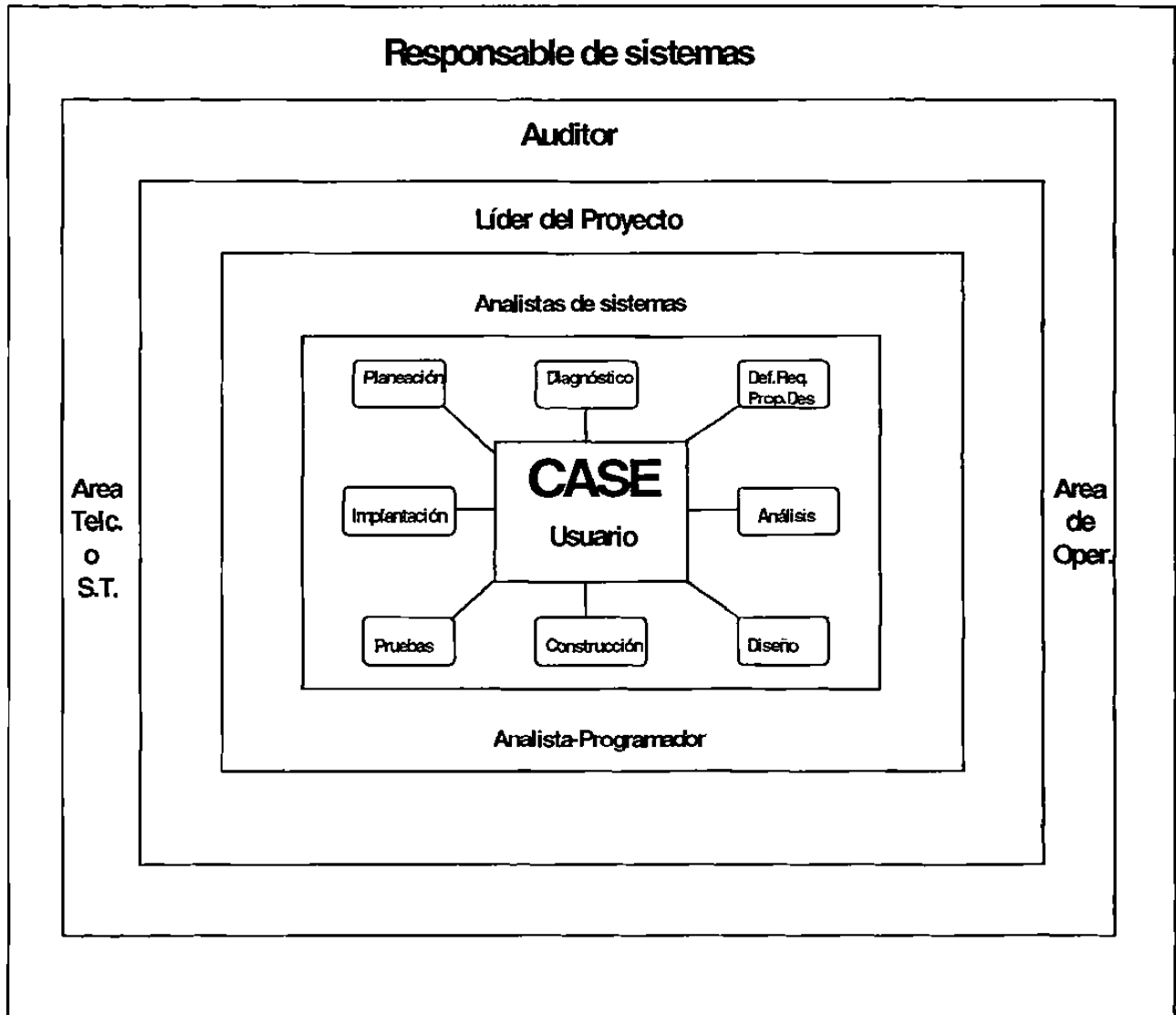
No deberán de aceptarse requerimientos de baja calidad o tecnología deficiente, ya que al aceptarlos el sistema carecerá de calidad incurriendo en altos costos por mantenimientos excesivos.

m) Reconocimiento.

Podemos establecer un sistema de premiación para reconocer a aquellos que logren sus objetivos o que su desempeño haya sido sobresaliente. Esto puede hacerse a través de felicitaciones en público o asignación a proyecto más retadores, promociones, etc.

n) Involucramiento total.

Para que se dé la calidad en el departamento de sistemas, toda la empresa deberá estar comprometida en el logro de la calidad siguiendo un programa de calidad previamente establecido, ya que de no ser así, puede ser que se fomente una resistencia al cambio por parte del personal de sistemas.



6. GUIA Y ESTRATEGIAS PARA DESARROLLAR E IMPLANTAR UN SISTEMA CON CALIDAD.

A través de esta guía la cual hace referencia al proceso de desarrollo de sistemas ideal, se ha decidido incluir las mejores técnicas (prototipo, modelación, JAD entre otras) y herramientas como CASE que aseguran no sólo el desarrollo de una aplicación de la manera más rápida, sino la participación del usuario además de un control en todo el proceso, logrando así un sistema con calidad.

Son muchas tecnologías las que se van a incorporar, varias ya conocidas y dominadas por algunos y otras como CASE que requerirán de un modelo o enfoque para implantarse como se vio en el capítulo 4.

Es importante considerar que para que esta guía tenga el éxito que se espera, las tecnologías incorporadas deberán asimilarse lo más pronto posible, siguiendo para ello algún proceso y contando con el apoyo de la alta administración; también se buscará lograr el entendimiento de esta guía y lo más importante es que las personas deberán estar convencidas de sus beneficios.

Para adoptarla se recomienda:

- Proporcionar un entrenamiento.
- Empezar a practicar con ella en proyectos pilotos.
- Mejorar la eficiencia con su uso.
- Ya una vez dominada será ejecutada sin ningún esfuerzo logrando excelentes resultados.

Acompañando a esta guía estarán por lo tanto, las posibles causas, que no deseamos que ocurran y sus cursos de acción para su posterior solución, esto será parte de la implementación del modelo de prevención indicado en el punto 5.7 de este proyecto.

Puede considerarse que suceda alguna causa y su curso de acción no esté contemplado, para lo cual deberá analizar cuáles son las causas que ocasionan determinado síntoma, para posteriormente determinar el curso de acción para su solución e implementarlo, actualizando esta guía para fomentar así, la mejora en futuros desarrollos.

Deberá considerar que todo cambio implicará una resistencia para lo cual tendrá que emplear los conceptos de administración de cambio, para lograr disminuir la resistencia cuando se haya conseguido aumentar el grado de aceptación.

Además sus miembros del equipo de desarrollo deberán de tomar en cuenta la experiencia de los:

- Analistas del área de soporte técnico o de telecomunicaciones según sea el caso, los cuales evaluarán el impacto que tendrá el futuro volumen de transacciones en relación al tiempo de respuesta de las terminales y al rendimiento que pueda tener la red, además podrán recomendar procesos para poder balancear la red.

- Analistas de operación, que ayudarán a definir los requerimientos tanto de memoria como de hardware, evaluarán el impacto de el nuevo volumen de transacciones en relación a los procesos que ya se ejecutan en ciertos períodos de tiempo, pudiendo además sugerir estructuras de los archivos de datos y formas de acceso para poder lograr una mayor eficiencia desde el punto de vista operacional, estimando el uso quizás de cintas y de discos indicando las características de desempeño máximo y normal, recomendando además procesos de recuperación y respaldo.

- Auditores, quienes dada su experiencia y conocimiento en :

- + Funciones del negocio.
- + Metodologías de desarrollo de sistemas.
- + Estándares para documentación.
- + Planeación de sistemas.
- + Administración de proyectos.
- + Etc.

Deberán sugerir técnicas de control o guías para cada fase compararlas con las que se han estado utilizando recomendando controles adicionales para fortalecer el medio ambiente de desarrollo. Siendo estos controles preventivos o detectivos y ya no correctivos. Su labor no será de revisar algo que ya esta hecho, ya que se realizarán las cosas bien a la primera sino que asesore al personal de desarrollo de sistemas para mejorar cada vez el proceso. ¿Por qué tener gente extra revisando lo que otros deben hacer bien?, esto resultaría muy costoso.

a) Planeación de Sistemas:

Actividades:

- Definición de objetivos.

Deberán de definirse los objetivos a nivel departamental, asimismo en este proceso deberán de participar los gerentes de los diferentes departamentos así como del área de sistemas, no olvide que los objetivos deberán ser específicos cuidando que sean realizables (no imposibles), concretos (claros), armonizados (deseables) y cuantificables (medibles) y lo cuales deberán estar relacionados de algún modo con los de la empresa.

- Deberán definirse los proyectos que apoyen los objetivos, considerando los proyectos generales del departamento y las solicitudes de los usuarios.

Es importante mencionar que estas solicitudes deberán de entregarse previo a la junta de planeación de sistemas y las cuales deberán de estar completas con respecto a la información requerida. Para ello esta solicitud elaborada por el usuario y revisada junto con él por un elemento del área de sistemas deberá estar apoyada por un anteproyecto que indique el objetivo, beneficios, tecnología propuesta, alcances, así como las actividades relevantes por subsistema que forman parte de él.

- Asignar prioridades, estas deberán ser asignadas por los gerentes o jefes del área(s)(clientes) involucrados con los sistemas. Utilice alguna técnica para darles prioridad.

- Asignación de recursos y fechas a los proyectos aceptados, esto deberá de registrarse haciendo uso de la herramienta CASE que cuenta con la opción de administración de proyectos. Esa fecha a su vez deberá ser considerada posteriormente en el plan detallado del proyecto para definir a su vez fechas y recursos de las actividades de las fases del desarrollo del nuevo sistema.

El apoyo de una herramienta CASE será muy importante ya que a través de ella se podrá establecer una estructura de trabajo (actividades), asignar tareas, tiempos y estimar costos utilizando el modelo de COCOMO. Estas herramientas además nos permiten manejar métodos como PERT-CPM y Gantt para controlar el avance del proyecto. Gracias a esta herramienta podremos:

- + Desarrollar y modificar con facilidad los diagramas de barras (Gantt) y de flujo (PERT-CPM), siempre resaltando este último el camino crítico.

- + Traducir los plazos del proyecto a fechas del calendario real, es decir tomando en cuenta los días festivos y vacaciones.

- + Con los diagramas de flujo que podremos construir lograremos:

- * Apreciar con claridad todas las actividades, los eventos y las estimaciones de tiempo.

- * Determinar fácilmente el tiempo en que puede realizarse el proyecto.

- * Coordinar todas las actividades.
- * Realizar análisis del tipo ¿qué puede salirnos mal? determinando así las consecuencias de una demora de cualquier actividad.

+ Y con los diagramas de Gantt podremos:

- * Seguir y controlar la marcha del proyecto y suministrar una retroalimentación importante al equipo de desarrollo que interviene en el proyecto.
- * Reflejar el presupuesto económico o las necesidades de personal o de equipo, nos ayudaría a evitar la sobre explotación de recursos sobre todo si van a ser compartidos en varios proyectos.
- * También estos diagramas nos van a ser de mucha ayuda para realizar el seguimiento y control de los gastos reales o de la utilización real de los recursos.
- * También es posible llevar a cabo análisis ¿Qué ocurriría si?, dibujándose automáticamente al introducir los datos la gráfica y así realizar los análisis para tomar la mejor decisión.

+ Con esta herramienta se podrá recuperar los plazos de un proyecto que se ha retrasado. Ya que si se observa algún margen de deslizamiento en un camino determinado se pueden trasladar los recursos sobrantes a las actividades del camino crítico mostradas por PERT, con el fin de agilizarlas.

- A final de cuentas si usted utiliza esta herramienta permitirá que el proyecto acabe dentro del plazo, del presupuesto y dentro de las especificaciones de calidad.

Productos terminados:

- Objetivos del departamento.
- Solicitudes de servicio que cumple con los requerimientos del departamento de sistemas (ver Anexo C).
- Anteproyectos de sistemas.(Información detallada en relación al sistema como objetivos, beneficios, tecnología propuesta, actividades relevantes por subsistemas y alcances).
- Asignación de fechas y recursos para realizar el diagnóstico para cada proyecto.
- Plan de sistemas.

Involucrados:

- Usuarios.
- Líder del proyecto.

- Responsable de sistemas.
- Auditor.

Responsables:

- Responsable de sistemas.

Técnicas y herramientas:

- Software administrador de proyectos (por ejemplo: Project Management).
- CASE soportando la parte de administración de proyectos.
- Técnica para dar prioridad a los proyectos. (Considerando las prioridades proporcionadas por el usuario).
- PERT-CPM o diagrama de flechas, GANTT.
- Documentación.
- Diagrama marcial, diagrama de afinidad, diagrama de árbol, diagrama de proceso de decisiones programadas, diagrama de relación, diagrama de pareto, hoja de verificación.

Causas que ocasionan una deficiente planeación:

- No se lleva a cabo la petición de un servicio a través de una solicitud formal.
- Usuario desmotivado por no cumplirle antes.
- Rechazo de solicitudes después de retenerla por varios días, ya que no cumple con las especificaciones requeridas por el departamento de sistemas. Información incompleta.
- Después de retener la solicitud de servicio por un tiempo éste no se lleva a cabo por que no es de alta prioridad, indicándole al usuario que no se realizará su proyecto.
- Por carecer de una planeación o por su mala práctica se desperdician recursos y muchas veces los proyectos que se planean no contribuyen a los objetivos del negocio.
- No se cuenta o no se asigna al personal idóneo con respecto al área a la que pertenece el proyecto a desarrollar.

Cursos de acción a seguir:

- Definir como política del departamento de sistemas que para dar servicio a un requerimiento o una necesidad del usuario, deberá realizarse una petición formal a través de una solicitud por escrito y firmada, presentándola además antes o en la reunión que se realiza para llevar a cabo la planeación de sistemas.

- Si la propuesta del usuario contribuye a los objetivos del negocio y apoya las estrategias de la empresa su proyecto será tomado en cuenta según la prioridad que el mismo usuario le dé atendándolo en base a los recursos con los que se cuente y a fechas asignadas.
- Si la información de la solicitud está incompleta o no es clara, planear una entrevista con el usuario para aclarar con él las dudas, de tal que la solicitud pueda ser completada.
- La recepción de solicitudes deberá realizarse en un período previo a la planeación estratégica de sistemas, de no ser así, las solicitudes que se reciban posteriormente podrán permutarse si el área en cuestión tiene asignado otro proyecto ya planeado y donde el tiempo y los recursos destinados a ese proyecto sean semejantes al nuevo.
- Deberá de manejarse una técnica para dar prioridad a los proyectos considerando ante todo los objetivos estratégicos de la empresa. Consultar y pedir opiniones antes de tomar una decisión.
- Dar una retroalimentación al usuario independientemente de cuál sea el resultado de la evaluación de su proyecto, (Aceptado alta prioridad, baja prioridad, o rechazado). Indicar las razones por su rechazo. Esto deberá de a través de un documento formal anexado a la solicitud de servicio, este documento deberá ir firmado por la persona que atendió la solicitud.
- Se deberán considerar los avances tecnológicos para mejorar este proceso de planeación, y registrarse todos aquellos proyectos en un administrador de proyectos con el fin de darles un seguimiento adecuado. Se registrará el nombre del proyecto y el responsable del mismo. Si no cuenta con la herramienta justifíquela haciendo una evaluación y preséntela en la siguiente etapa.

b) Diagnóstico:

Deberá de analizarse la necesidad que ha sido presentada por el usuario, llevando a cabo un análisis preliminar del sistema actual, con el fin de determinar cuál es la problemática u área de oportunidad que se presenta. Se deberá de investigar el funcionamiento de todas las partes del sistema actual conociendo :entradas, procesos, salidas, tecnología que utiliza, áreas y gente involucrada. El tiempo que se le dedique a esta fase será determinado por el líder del proyecto, este dependerá mucho de la magnitud del proyecto y de la cantidad de recursos que éste disponga, lo que es importante recalcar es que esta fase siendo la base del desarrollo del sistema requerirá de más tiempo junto con el análisis que una fase de diseño, construcción y pruebas. Recuerde que si no se entiende el problema no será posible encontrar la solución.

Actividades:

- El líder del proyecto seleccionará a las personas idóneas para llevar a cabo el diagnóstico. Estas personas deberán de ser analistas con amplia experiencia práctica en el campo de los negocios y con las siguientes habilidades:
 - + Guiar y ayudar a descubrir los problemas.
 - + Que sepa trabajar en equipo.
 - + Saber escuchar.
 - + Expresar ideas claras y en forma breve.
 - + Aceptar puntos de vista de otros.
 - + Vendedor de ideas.
 - + Ser observador y detectar lo que está oculto y que no ha salido a flote.
 - + Formular las preguntas adecuadas para obtener información relevante.
 - + Inspirar confianza en la gente y animarla a que digan lo que realmente piensan.
 - + Manejar técnicas y herramientas para obtener información relevante.
 - + Concretizar y sintetizar describiendo con objetividad los problemas.

Siendo todo esto indispensable si se desea que esta fase sea exitosa dando el diagnóstico correcto, reconociendo así los problemas y sus causas.

- El proceso a seguir para reconocer cuáles son los problemas y sus causas, será:
 - + Realizar un plan para llevar a cabo la recopilación de información.
 - + Seleccionar la(s) técnicas para efectuar la recopilación.

Cuestionarios: los cuales representan entrevistas autoaplicadas. A través de ellos es posible cuantificar las respuestas, utilizándolo sobre todo cuando la muestra de personas es grande.

Entrevistas: A través de la cual podremos descubrir la forma en que la organización funciona, es considerada como una fuente rica de información.

Observación: Por medio de esta podremos observar el comportamiento a medida que este ocurre, obteniendo la realidad.

- + Diseñar el cuestionario, las preguntas a realizar en la entrevista, así como definir que observar y a quienes según sea el caso.
- + Llevar a cabo la recopilación de información.
- + Una vez recopilada la información construya un modelo (de procesos y de datos) en una sesión de lluvia de ideas (*brainstorming*), auxiliado por una herramienta CASE que soporte esta actividad, utilizando técnicas de análisis estructurado, logrando así representar la situación real para poderla analizar y determinar tanto los problemas como sus causas. La herramienta CASE le será de mucha utilidad para construir un modelo de alto nivel de la empresa, así como el modelo conceptual del sistema utilizando el diagramador de procesos y de flujo de datos, ayudando al analista(s) a determinar las redundancias que existen en el proceso modelado.
- + Auxíliese además de estadísticas representadas por gráficas, basándose en las herramientas estadísticas y administrativas de calidad (Anexos).
- + Analice y sintetice la información generando un diagnóstico parcial.
- + Presente al usuario los resultados mostrándole de ser posible los modelos de procesos creados.
- + Si el cliente acepta los resultados, este podrá a su vez retroalimentar considerando esta información, en un proceso de consolidación de información. Quizás se requiera una modificación a los diagramas ya elaborados esto no será problema si estos puedan ser corregidos inmediatamente al utilizar la herramienta CASE.
- + Consolide la información generando así un diagnóstico general indicando :
 - * Los objetivos de la empresa.
 - * La estructura actual de la organización , sus productos y servicios, áreas afectadas sus clientes, las políticas establecidas, información sobre sus métodos y procedimientos, la descripción de puestos y toda la información relacionada con el sistema actual (procesos, información tanto de entrada como de salida).
- + Lista de problemas y sus causas así como el objetivo del proyecto a realizar como una solución posible el cual deberá de ser verificado por el usuario, no olvide que este objetivo deberá ser realista, claro y medible.

Productos terminados:

- Datos generales de la empresa
- Objetivos de la empresa.
- Estructura organizacional. Descripción de puestos.
- Lista de usuarios y áreas afectadas.

- Políticas establecidas.
- Manuales de métodos y procedimientos.
- Información relacionada con el sistema actual como :
 - + Software y hardware en el que se desarrolla el sistema.
 - + Escenario de seguridad que rodea al sistema.
 - + Información que requiere y genera el sistema.
 - + Procesos que son cubiertos por el sistema.
- Modelo del negocio.
- Modelación de procesos y de datos del sistema actual.
- Estadísticas haciendo uso de las herramientas de calidad.
- Lista de problemas y sus causas.
- Objetivo del proyecto.
- Plan de recopilación de información.
- Cuestionarios, resultados de las entrevistas y de la observación.

Involucrados:

- Usuario.
- Analista.
- Auditor.
- Líder del proyecto.
- Areas de soporte al sistema.
- Proveedores/Clientes relacionados con el sistema.

Responsables

- Líder del proyecto.
- Analista.

Técnicas y herramientas:

- Entrevistas.
- Cuestionarios.
- Observaciones.
- Causa-efecto.
- Análisis estructurado.
- Documentación.
- CASE (Middle-CASE, Lower-CASE).

- Brainstorming.
- Diagrama de seis palabras.
- Pareto.
- Estratificación.
- Diagrama de dispersión.
- Gráfica de control.
- Diagrama de relaciones.
- Diagrama de árbol.
- Hojas de chequeo.
- Diagrama de afinidad.
- Diagrama matricial.
- Documentación.

Causas que ocasionan un diagnóstico deficiente:

- No se involucra al usuario.
- No hay un compromiso por parte del usuario, piensa que pierde su tiempo.
- El usuario tiene dificultad para comunicarse.
- El analista no conoce o no utiliza las técnicas y herramientas.
- El analista no entiende el problema.
- No se realiza un estudio del sistema actual.
- Inconsistencia entre la problemática especificada por el usuario y la determinada por el analista de sistemas.
- Objetivos no claros, cuantificables o controlables.
- No existe la documentación del sistema anterior.

Cursos de acción a seguir:

- *El usuario es una pieza clave para entender el problema y conocer todos los procesos e información que éstos requieren y producen. Deberá siempre involucrarse si se desean obtener excelentes resultados.*

- El usuario a veces no se compromete por temor a ser desplazado o porque se resiste al los cambios que puedan ocurrir. Hay que convencer al usuario de que se le ayudará a realizar mejor su trabajo para que así éste pueda desempeñar otras funciones.
- Se requerirá de habilidad por parte del analista para obtener la información que se desea cuando el usuario tiene problemas para comunicarse. Se le recomienda diseñar un cuestionario afín a la problemática que lleve al analista a localizar, y enfocar en forma correcta y concreta el problema existente.
- Capacitación en el uso de las metodologías, técnicas y herramientas para que conozca ¿qué son? ¿Cuándo y cómo utilizarlas? además de conocer los beneficios que con ellas se pueden lograr.
- Para que el analista entienda mejor la problemática que se presenta en una determinada área, deberá estudiar antes aquellos procesos que puedan estar relacionados con el área a investigar (indicada en la solicitud de servicio), y además deberá apoyarse con bibliografía para entender aquellos tecnicismos que no se conocen o que en un momento dado no hayan sido bien explicados.
- El analista deberá conocer cual es la técnica (entrevistas, cuestionarios, observación) más adecuada para recopilar la mayor cantidad de información.
- En un caso dado que la problemática entre ambas partes (usuario y analista) tenga un enfoque muy diferente, llevar a cabo sesiones extras para resolver esas diferencias y aclarar dudas, buscando que el enfoque de los problemas sea uniforme y realista.
- Muchas veces los objetivos no son claros porque el analista no ha comprendido ni la problemática, ni los procesos involucrados en ellos. Si es necesario deben aclararse nuevamente las dudas teniendo breves sesiones con los usuarios de las áreas involucradas.
- Si no existe la documentación se llevará más tiempo el realizar un análisis del sistema actual, sin embargo si éste existiera no hay que olvidar que se debe registrar en una herramienta CASE, o si ya existe, podrá reutilizarse para elaborar la nueva modelación de datos y procesos para el nuevo sistema. En dado caso que sólo se cuente con el diseño físico de la base de datos, puede aplicarse la ingeniería en reversa con CASE para generar así el modelo entidad-relación deseado.

c) Definición de requerimientos y desarrollo de la propuesta.

Definir y documentar los requerimientos que se pretenden obtener por medio del sistema. Tal que se determine:

- Quién es el que necesita la información.
- Qué información necesita, cuándo y dónde.

Logrando en base a esos requerimientos desarrollar la propuesta del proyecto a presentarse para su posible aceptación.

Actividades a seguir:

- El analista junto con el usuario deberá de definir y documentar los requerimientos de:

Funcionalidad.
Desempeño.
Documentación.
Seguridad.
Calidad.
Datos.
Medio ambiente.
Organizacionales.

(a) Tomando como base la información recopilada en la etapa anterior así como los diagramas de flujo de datos y entidad-relación, y quizás información de entrevistas y/o cuestionarios extras, podremos determinar cuales son los requerimientos, y (b) construir un prototipo el cual esté basado en esos requerimientos encontrados inicialmente. (c) Una vez construido el prototipo auxiliado por una herramienta CASE, el usuario podrá interactuar con él de tal manera que se determinen nuevos requerimientos o cambios en los diseños y formatos de entrada y salida. (d) Posteriormente el analista realizará las modificaciones pertinentes repitiendo el proceso (c) y (d) hasta que el usuario quede satisfecho.

Como construir el prototipo:

Utilizando herramientas CASE:

- + Podrá definir las entidades o archivos a utilizarse en las pantallas y los reportes, esto se hará si no han sido definidos en el modelo entidad-relación del sistema actual. (Considere los puntos de vista del analista de operación).
- + Crear las pantallas y reportes definiendo títulos, encabezados y campos previamente definidos, si desea que todos los campos estén incluidos solo basta hacer referencia al archivo o entidad generándose estos automáticamente en el reporte o en la pantalla.
- + Crear los menús que harán referencia a esas pantallas y reportes.

- + *Crear el prototipo final* indicando la secuencia de menús, pantallas y reportes. soportado el prototipeo por CASE es posible construir y modificar el prototipo rápidamente ayudando a que el proyecto no se retrase de acuerdo a lo planeado, logrando una calidad e integridad de la información.

En dado caso que no tenga una herramienta CASE que soporte el prototipeo, puede hacer uso de un lenguaje 4GL o algún software de aplicación (DBASE, FoxBase, Access) utilizando sus generadores de pantallas y reportes. Será indispensable utilizar estas herramientas cuando necesitemos realizar cálculos especiales o manejar cierta lógica que muchas veces no es posible con las herramientas CASE. Para construir su prototipo con este tipo de software deberá:

- + Construir sus archivos.
- + Construir sus menús.
- + Construir la pantallas para dar entrada a información.
- + Construir los reportes.
- + Construir otras funciones quizás procesos batch, llamadas a lenguajes de tercera generación, cálculos matemáticos etc.

Y seguir el mismo proceso de verificar con el usuario y modificar el prototipo hasta que quede listo. Si la herramienta CASE que usted posea cuenta con un lenguaje 4GL o si le permite interfaces con algún otro lenguaje esto le ayudará a construir un prototipo que satisfaga las necesidades de sus usuarios.

La calidad que se logre obtener de los requerimientos se obtendrá si éstos llegan a refinarse a través del prototipo.

- Elaboración de la propuesta. Considere al elaborar la propuesta la información y restricciones que le pueda proporcionar el analista de operación y de telecomunicaciones.

En base a los requerimientos previamente especificados se deberá definir:

Objetivos particulares del proyecto. Deberán estar relacionados con los requerimientos definidos.

Alcances.

Limitaciones.

Diferentes alternativas para solucionar el problema.

- + *Mantenimiento.*
- + *Desarrollo de un nuevo sistema.*
- + *Compra del sistema.*
- + *Continuar con el sistema como está.*

Definir para cada alternativa:

- + *Factibilidad económica: Realizar un análisis costo-beneficio, (utilizando técnicas de comparación de costos y beneficios así como aquellas relacionadas con la estimación de costos), estimando el costo de operación del sistema actual, el costo de operación del sistema propuesto e indicar aquellos beneficios tanto tangibles como intangibles*

determinando de esta manera si los costos pueden compensarse por las ganancias a largo plazo.

*+ Factibilidad técnica: Determinar si los recursos técnicos como el hardware, software y el soporte técnico que se pueda tener satisfacen los requerimientos definidos.
De no ser así, determinar si la actual tecnología se necesita modernizar o si se requiere de una nueva, tomando en cuenta que esto pueda ser justificable financieramente.
Realizar una evaluación de hardware y software según se requiera seleccionando aquel que satisfaga más las necesidades.*

+ Factibilidad operativa. Ver si se cuenta con el personal capacitado para soportar el desarrollo de sistemas, de no ser así determinar si es viable capacitarlo.

+ Factibilidad de recursos humanos. Independientemente si se va a requerir de un nuevo sistema o de una modificación a uno ya existente hay que recordar que esto provocará en la mayoría de los casos una resistencia al cambio por parte de las personas afectadas directamente por el sistema. Es importante considerar si estas personas pueden obstaculizar el desarrollo o implementación del sistema, determinando el problema que representan, cambiando la resistencia que estas personas puedan tener basándose en la presentación de las ventajas que se pueden obtener por medio del sistema. A pesar de esto, puede persistir la resistencia y por lo tanto debe convencerse, o de ser necesario, presionar para lograr los fines que se requieran.

Realizar el plan general del proyecto indicando:

- + Actividades a realizar.
- + Responsables.
- + Fechas de inicio y terminación de cada actividad, considerando los compromisos de inicio del sistema por requerimientos externos.
- + Metodologías, técnicas, herramientas y estándares propuestos.
- + Plan de calidad.
- + Plan de documentación.
- + Plan de pruebas.
- + Plan de entrenamiento.
- + Plan de seguridad(contingencias y recuperación).

Haciendo uso de la herramienta CASE, deberán de registrarse las fases y las actividades para la realización de los diferentes proyectos propuestos, así como los responsables indicando la fecha de inicio y terminación, las horas a dedicarle y el costo total por cada una de las actividades. Cabe mencionar que ésta información no podrá ser registrada si el proyecto no ha sido dado de alta en el sistema.

Registrando esta información el administrador del proyecto podrá consultar el avance del mismo conociendo el estado del proyecto e identificando las desviaciones del plan para llevar a cabo las acciones adecuadas con el fin de asegurar los resultados deseados.

- Preparar presentación de la propuesta.
- Presentación de la propuesta (Vender el proyecto).
- Rechazo o aceptación del proyecto por parte del usuario o de la gerencia a la que éste pertenece y del responsable de sistemas.

Productos terminados:

- Documentación conteniendo los requerimientos.
- Propuesta de desarrollo.
- Plan del proyecto.
- Documento firmado de aceptación del proyecto.

Involucrados:

- Gerencia a la que pertenece el usuario.
- Usuario(s).
- Analista de sistemas, de operación y de telecomunicaciones.
- Líder de proyectos.
- Responsable de sistemas.
- Auditor.

Responsables:

- Líder del proyecto.
- Analista.

Técnicas y herramientas a utilizar:

- Un software administrador de proyectos. (por ejemplo: Project Management).
- GANTT.
- PERT-CPM o diagrama de flechas, para apoyar la planeación y control de proyectos si no cuenta con la herramienta CASE.
- CASE con soporte a la administración de proyectos (Categoría Upper-CASE).
- Diagrama de pareto.
- Métodos de estimación de costos (COCOMO).

- Técnicas de comparación de costos y beneficios:
 - + Análisis del punto de equilibrio: Utilizarse cuando se necesite justificación por costo y no por los beneficios.
 - + Análisis del flujo de efectivo: hacer uso de éste cuando el proyecto es muy costoso.
 - + Valor presente: Utilizarse cuando el período de inversión es largo o si el costo del financiamiento es elevado.
 - + Retorno de inversión: Se recomienda su uso si los beneficios tangibles mejoran.

- Utilizar técnicas para recopilación de información (requerimientos).
 - + Cuestionarios.
 - + Entrevistas.

- Técnicas para identificar los requerimientos.
 - + Prototipeo.
 - + JAD.

- Documentación.

Causas que ocasionan una definición de requerimientos y una propuesta de desarrollo deficiente:

- Que no se realice el análisis de requerimientos.
- No se conocen las técnicas, herramientas o metodologías para realizar esta actividad.
- Equipo inadecuado para el uso de las herramientas, no soporta el software.
- Demora en la aceptación de proyectos.
- Requerimientos incompletos.
- Requerimientos no viables.
- Persisten los problemas con el usuario para obtener la información.
- Objetivos no claros, cuantificables o controlables.

Cursos de acción:

- Para obtener la información acerca de los requerimientos no se debe olvidar utilizar las técnicas como la entrevista o cuestionarios.

Si se ha decidido realizar una entrevista se le recomienda que lleven a cabo los siguientes pasos:

- a) Lectura de antecedentes.
- b) Establecer el objetivo de la entrevista.
- c) Seleccionar a los entrevistados.
- d) Preparación del entrevistado: Es importante hacer sentir al entrevistado que la información que proporcione será de gran utilidad para poder lograr la calidad en el proyecto a desarrollar. Además de esto, deberá de obtenerse una cita y el entrevistador le deberá indicar cuáles serán los diferentes puntos a tratar.
- e) Estructurar las preguntas

Además de tomar notas para documentar la entrevista, puede apoyarse de una grabadora para registrar completamente todo lo que mencione el entrevistado, además con el uso de la grabadora puede realizar más preguntas y obtener así más información. Salvo que el entrevistado no este de acuerdo en el uso de la grabadora hay que aceptarlo.

Si se ha decidido aplicar un cuestionario deben considerarse los siguientes puntos que le indican la conveniencia en su uso:

- a) Las personas a quienes necesita interrogar se encuentran localizadas en diferentes áreas.
- b) Existe una gran cantidad de usuarios, es indispensable conocer sus necesidades.
- c) Es necesario llevar a cabo un estudio exploratorio.

- Vender la idea acerca de los beneficios que se logran con el uso de las metodologías, técnicas o herramientas.

- A través de una actualización continua en tecnología de información, evalúelos y presente los resultados a la alta dirección para su aprobación.

- Se recomienda construir un prototipo con el cual el usuario pueda trabajar para determinar si los requerimientos obtenidos son los que se desean y además detallarlos. Esto podrá ser soportado por una herramienta CASE(Categoría Lower-CASE) con el fin de mejorar la comunicación con los usuarios cuando sobre todo los requerimientos no son claros, detallando así los requerimientos en un tiempo más corto, como resultado de la interacción con el prototipo. No es recomendable para sistemas de tipo batch o sistemas que incluyen procesos de cálculo complejos.

- Deberá adoptarse como una política del departamento, que todo proyecto deba ser presentado con los lineamientos anteriormente mencionados (Objetivos, alcances, limitaciones, costo-beneficio, etc.) Es muy importante que el costo y beneficio esté presente, ya que de no ser así, no se podrá tomar una decisión para continuar con el desarrollo del proyecto. Debe considerarse los siguientes puntos para realizar un análisis costo-beneficio.

- a) Identificar los costos y los beneficios tanto tangibles como intangibles.
- b) Llevar a cabo estimaciones para los costos y beneficios tangibles.
- c) Y por último utilizar una de las metodologías anteriormente mencionadas para llevar a cabo la comparación de los costos y los beneficios.

No olvide que a lo largo de todo el proyecto haciendo uso de la herramienta CASE que soporta la administración de proyectos los miembros del equipo de desarrollo registrarán todas aquellas actividades que se vayan terminando, tal que el líder del proyecto pueda verificar el avance del proyecto, realizando los ajustes necesarios con el fin de lograr un sistema dentro del tiempo y del presupuesto previsto. Los cambios tendrán que ser informados al equipo de desarrollo de manera clara, oportuna y comprensiva; estos podrán ser dados a conocer fácilmente teniendo acceso a la información que maneja la herramienta CASE la cual se encuentra almacenada y actualizada en su repositorio. Asimismo el usuario podrá realizar consultas teniendo acceso a la información almacenada en el repositorio de CASE, conociendo así el grado de avance de su proyecto.

d) Análisis:

Actividades:

d.1) Modelación de datos.

El modelo de datos que se definirá será la base del desarrollo del sistema .

Debe quedar claro que el éxito de esta actividad dependerá no sólo de que el usuario esté involucrado en cada paso en el proceso de la modelación, sino de que entienda perfectamente el modelo y lo revise, respondiendo el analista rápidamente a las sugerencias asistido por una herramienta CASE en sesiones JAD y además deberá de buscarse que el modelo esté normalizado.

Para construir el modelo o diagrama entidad-relación se tomará como referencia los preceptos de Chen. Con este diagrama se mostrarán todas las entidades y sus datos indicando además la manera como estas se relacionan.

Al construir este modelo estaremos separando el diseño conceptual de datos del diseño físico para posteriormente una vez elegida la base de datos donde vamos a implementar el modelo este podrá convertirse en un diseño físico.

Haciendo uso del editor que posee la herramienta CASE usted podrá hacer referencia al modelo de datos previamente creado en la fase de diagnóstico, si desea modificarlo, o podrá crear uno nuevo tal que las entidades sean representadas por rectángulos, las relaciones por rombos y mediante las líneas pueda conectar los conjuntos de entidades al conjunto de relaciones, mostrando además la cardinalidad.

Su construcción será muy fácil ya que a través de una paleta que maneja la herramienta se muestran los diferentes símbolos tal que usted podrá ir seleccionando el símbolo que desee, así hasta terminar de construir su diagrama.

Además para definir los atributos que compongan a cada entidad deberá indicar su nombre, su formato y si es o no llave. Gracias a la herramienta CASE podremos darnos cuenta de aquellas asociaciones no definidas basándose en las llaves definidas como foreign, reportando además las redundancias. Todo el modelo quedará almacenado en un repositorio para su uso posterior como será su normalización, su conversión a un esquema deseado o así mismo podrá ser accesado para hacer referencia a él cuando se realice modelo conceptual del sistema.

He aquí los pasos para construir el modelo utilizando la herramienta CASE la cual soporta la técnica de análisis estructurado, involucre al analista de operación y de telecomunicaciones:

- Se deberán de tomar en cuenta los requerimientos de información definidos en la etapa anterior, considere además el modelo de datos ya construido del sistema actual, almacenado en el repositorio.
- Identifique cuales serán las entidades a formar parte del nuevo esquema conceptual, podrá modificar el modelo ya creado o crear uno nuevo, considere que para la realización del

prototipo ya fueron definidas las entidades o archivos basta hacer referencia a ellas al estar formando el modelo entidad-relación.

- Establecer la relación entre las entidades con los rombos e indique la cardinalidad.
- Normalice el modelo. La herramienta CASE podrá normalizar el modelo automáticamente hasta la tercera forma normal, asegurando así la corrección, exactitud, eficiencia y estabilidad. Logrando además no sólo que el conjunto de entidades sean precisas y no redundantes sino que las relaciones complejas se dividan en simples.
- Cheque el modelo con los usuarios.
Dado que este tipo de modelos es fácil de entender ya que se acerca más a la realidad es necesario que este sea verificado con el usuario, el analista de operación y de telecomunicaciones, para determinar que cumple con sus requerimientos.
- Resolver las discrepancias.
- Modificar el modelo.
- Repetir las actividades a partir de la normalización hasta que el usuario esté de acuerdo. Se recomienda realizar esto en una sesión JAD.
- Convierta el diagrama entidad-relación o esquema conceptual a un esquema DBMS si lo desea o al manejador de archivos a utilizar, pudiéndolo transferir a donde resida el software DBMS o manejador de archivos.

Que quede claro que las estructuras de la base de datos podrán cambiar en un futuro pero el objetivo de esta modelación será el de minimizar esos cambios. Las estructuras en el negocio deben ser estables, mientras que los procesos tienden a cambiar constantemente.

d.2) Modelación de procesos.

Es importante determinar cuáles serían los procesos que utilizarían todas las entidades definidas en el modelo de datos, generado en la actividad anterior.

Actividades:

La herramienta CASE a través de su módulo modelador de diagramas de flujo de datos (DFD) y de procesos podrá soportar todas las siguientes actividades a realizar tal que la herramienta a través de su poderoso editor gráfico nos permitirá crear estos diagramas utilizando los símbolos básicos que representan los procesos, flujo de datos, almacenamiento de datos, y entidades externas, siendo los procesos explotados a múltiples niveles. Lo más importante que se obtiene al utilizar esta herramienta es que por la fácil manipulación de símbolos y su fácil uso se logra una máxima productividad, siendo la comprobación de errores de sintaxis y tipo, así como la verificación de la consistencia y el balanceo una de las funciones más importantes de la herramienta asegurando así la integridad y exactitud de los diagramas de flujo de datos.

Todos los diagramas serán almacenados en un repositorio pudiendo ser reutilizados para futuros proyectos ya sea en su forma parcial o total.

- Tome como referencia antes de construir el nuevo diagrama de flujo de datos, información sobre los procesos involucrados así como otras soluciones puestas en práctica por otras empresas.
- Realice la descomposición del proceso en subprocesos hasta llegar al detalle generando así los diagramas de procesos.
- Asignar nombres a los procesos de preferencia la composición del nombre será (verbo+ nombre del objeto) .
- Definir la relación entre los procesos y realizar un diagrama de dependencia de procesos.
- Definir una matriz en la cual se establezca la relación entre procesos y entidades previamente ya definidas y validadas y verificadas por el usuario.
- Realizar el diagrama de contexto.
- Realizar el diagrama de flujo de datos tomando en cuenta la matriz previamente definida. Generando diagramas hasta el nivel más bajo que se requiera, de esta manera estará utilizando la técnica top-down la cual consiste en describir más detalladamente las burbujas o procesos del DFD cero. Puede hacer referencia al DFD creado previamente en la etapa de diagnóstico.
Una de las mayores ventajas al estar utilizando herramientas CASE para la producción de estos diagramas es que en las sesiones JAD y llevando a cabo una lluvia de ideas pueden llegar a esbozarse varias opciones de solución utilizando el diagrama de flujo de datos y estando presente él o los usuarios puede llegarse a la mejor solución, no olvide considerar las políticas de la organización. Recuerde que ya no se tratara de automatizar lo que se estaba realizando si no de ver cual opción nos ayudará a mejorar y a la vez a reducir costos, sin olvidar el cumplimiento de los requerimientos antes definidos.
- No olvide numerar cada uno de los procesos que están definidos en el diagrama de flujo de datos, indicando el flujo de datos entre ellos.
- Redibujar el diagrama de flujo de datos para llevar a cabo correcciones como resultado de las revisiones de éste con el usuario, con analistas de operación y telecomunicaciones.
- Realizar el diccionario de datos. (Describir los flujos de datos así como los eventos tanto del modelo interno como del externo). Esto será generado automáticamente por la herramienta CASE.
- Verificar el balanceo y la consistencia , las herramientas CASE tienen una opción para llevarse a cabo esta actividad automáticamente, esta opción puede activarse o desactivarse según lo requiera, evitando:
 - + Evitar que los procesos sean infinitos (que no tengan una salida).
 - + Evitar procesos de generación espontánea (que no tengan entradas).
 - + Los datos de entrada y salida en los procesos hijos deben estar también en sus procesos padre.
 - + La información que se defina en el diagrama de flujo de datos debe encontrarse en el diccionario de datos.
- Realizar las miniespecificaciones. Esta son aplicables a aquellos procesos que no se descomponen en más subniveles de detalle, estas describen las actividades que se realizan en

los diagramas de flujo de datos para convertir las entradas y salidas, utilizar para ello el español estructurado. Algunas herramientas no tienen la opción de manejar el español, así que tendrá que trabajar en inglés.

- Realice los diagramas de transición de estados. Para representar la conducta del sistema, cuando este sea un sistema de tiempo real.

d.3) Definir la información relevante tanto de entrada como de salida para cada proceso. Existen templates que maneja la herramienta CASE para definir la documentación de sus pantallas reportes, formas y archivos. De hecho al generar sus pantallas, reportes y archivos toda la documentación relacionada a estos estará almacenada en un repositorio pudiéndola imprimir en el momento que se desee, estando siempre actualizada.

La información recomendable a definir será:

- Nombre, objetivo, tipo de forma (manual o por sistema), formato, fuente, frecuencia de uso, para formas.
- Nombre, objetivo, tipo de pantalla (actualización y consulta), frecuencia de uso, para pantallas.
- Nombre, objetivo, formato, fuente, frecuencia con que se genera, para reportes.
- Nombre, Objetivo, tipo de archivo, fuente, frecuencia de uso, para archivos

d.4) Actualización del plan.

Quizás se requiera de actualizar o cambiar el plan general, tal que se requiera de más recursos, como resultado del análisis.

d.5) Realizar la reevaluación de costos.

Si se requiere de más recursos para la realización de este proyecto, realice nuevamente estimaciones de costos, no olvide utilizar el método de COCOMO.

d.6) Reevaluar la justificación.

Dado los nuevos costos que no se habían tomado en cuenta será necesario actualizar el documento de análisis costo-beneficio empleando nuevamente técnicas de comparación de costos.

d.7) Obtener la aceptación del usuario o del cliente y del responsable de sistemas. Deberá analizarse el nuevo análisis costo-beneficio, si lo hay. El usuario determinará si se continúa o no con el proyecto, determinado si es rentable o no el proyecto y si los costos y beneficios no han cambiado. Deberá ser formalizado a través de un documento.

Productos terminados:

Toda la documentación relacionada con estos productos será generada automáticamente a través de la herramienta CASE, ya que ésta solo hará acceso al repositorio de información donde todos estos elementos se encuentran. Con excepción del análisis costo-beneficio y del documento formal de aceptación.

- Esquema conceptual (Modelo de datos de alto nivel).
- Modelo de datos de alto nivel normalizado.
- Esquema conceptual (Modelo de datos en un específico DBMS o en un manejador de archivos).
- Esquema interno (para el mismo DBMS o manejador de archivos).
- Diagrama de procesos (DP).
- Diagrama de dependencia de procesos (DDP).
- Matriz entidad-proceso (MEP).
- Defina el diagrama de contexto (Modelo externo)
- Diagrama de flujo de datos (DFD Modelo interno).
- Miniespecificaciones por cada proceso (ME).
- Diagrama de transición de estados del sistema (DTE).
- Diccionario de datos (DD).
- Toda la documentación relacionada con : archivos, pantallas, reportes y formas.
- Plan general actualizado.
- Análisis costo-beneficio actualizado.
- Documento final que indique la aceptación de usuario y del responsable de sistemas.

Involucrados:

- Líder del proyecto.
- Analista de sistemas, operación y de telecomunicaciones..
- Usuarios.
- Auditor.
- Operación.

Responsables:

- Líder del proyecto.
- Analista.

Técnicas y herramientas a utilizar:

- Entrevistas. Si no se realizan sesiones JAD.
- Análisis estructurado.
- Técnica de modelación.
- Técnica de normalización.
- Técnica Top-down.
- Software administrador de proyectos (por ejemplo Project Management).
- JAD.
- Case (Categoría Middle-CASE) soportando el análisis el cual siempre es estructurado..
- CASE soportando la administración de proyectos.
- PERT-CPM o diagrama de flechas.
- GANTT.
- Diagrama matricial, diagrama de relación, causa-efecto.
- Técnicas de comparación de costos.
- Estimación de costos (COCOMO).
- Documentación.

Causas que ocasionan un análisis deficiente:

- No se entienden los requerimientos especificados en la etapa anterior.
- No se cuenta con la información suficiente para identificar entidades y llevar a cabo la relación entre ellas y con los procesos.
- No se conocen, no se usan o no se saben utilizar las técnicas y herramientas. Muchas veces no se utilizan por que la herramienta no es clara y simple.
- Esquema conceptual está demasiado normalizado.
- Mala definición de índices.

Cursos de acción:

- Llevar a cabo entrevistas si la definición de requerimientos fué realizada por otras personas, para aclarar dudas.
- Cursos de capacitación apoyados por el responsable de sistemas.

- Determinar si la herramienta es la adecuada por medio de una evaluación, de no ser así justificar la compra de otra. Es indispensable para evitar la no realización de la documentación por lo tedioso que pueda implicar su ejecución a mano. Además muchas veces los cambios recomendados no se llevan a cabo si no se cuenta con una herramienta. Por ejemplo si trabaja con los diagramas de flujo podrá realizar:

- + Cambios continuamente.
- + Diagramas de flujo de datos a diferentes niveles.
- + Verificar la consistencia y el balanceo.

- Hay que determinar si es conveniente la normalización ya que algunas veces se disminuye el rendimiento en el sistema.

- Hay que verificar en el momento de realizar el diseño físico que los índices sean adecuados.

e) Diseño.

A través del diseño se podrá construir el modelo conceptual computacional del sistema necesario para su programación. Conociendo el problema y los requerimientos, se logra definir la solución computacional utilizando para ello técnicas y herramientas gráficas.

Actividades:

- Realizar el diseño lógico del sistema, considere la información e inclusive las posibles restricciones dadas a conocer por el analista de operación y de telecomunicaciones.

Por medio de la herramienta CASE podrán generarse automáticamente los diagramas estructurados accedendo al diagrama de flujo de datos (DFD) encontrado en el repositorio o bien si la herramienta no tiene esta poderosa función será a través del editor de la herramienta como tendrá que construir los diagramas estructurados.

Para ello deberá:

- + Definir la estructura general del sistema.
- + Para cada módulo utilice la técnica top-down para dividirlo en más módulos pequeños especificando el módulo al detalle que desee, tome como referencia el DFD ya creado. Será necesario indicar el flujo de información tanto de datos como de control en las llamadas de un módulo a otro.
Las herramientas CASE utilizan el rectángulo para representar el módulo, flechas con un círculo relleno o no relleno indicando el flujo de control o de información respectivamente añadiendo un identificador para darle un significado al flujo.
Debido a que la lógica de cada módulo no se especifica debe detallarse cada módulo indicando su nombre, la lista de parámetros de entrada y salida, el valor de retorno del módulo y su función.

CASE verificará la exactitud y eficacia del diseño estructurado indicando aquellas inconsistencias y eliminándolas.

Será indispensable para asegurar la calidad:

- + Utilizar técnicas de diseño estructurado soportado por la herramienta CASE.
- + Una vez construidos los diagramas estructurados determinar si existe un bajo grado de acoplamiento (parámetros que existen entre los módulos de todo el diagrama) asegurando así su reusabilidad, y el alto grado de cohesión (integración de las instrucciones) para asegurar su facilidad de cambios.
- + Al realizar el diseño la división realizada en módulos nos asegura un cumplimiento de los factores de calidad: flexibilidad, facilidad de prueba, portabilidad.

- Realice los casos prueba.

El diseñador deberá definir tanto los datos de entrada a cada uno de los programas a probarse así como las funciones del programa a ejecutarse con esos datos y la propuesta del resultado esperado con el fin de que sean tomados en cuenta al hora de realizar las pruebas para verificar y validar tanto los programas que conforman al sistema, como el propio sistema una vez ya integrados estos.

La herramienta CASE puede agilizar este proceso generándole los datos prueba tomando como base la estructura del programa, pero es hasta la siguiente etapa cuando puede utilizar la herramienta. Sin embargo las funciones a realizar y los resultados tendrían de todos modos que definirse.

- Crear opciones técnicas.

Las opciones tanto de hardware como de software deben haberse contemplado en el estudio de factibilidad. Si la decisión no ha sido tomada, deberá de tomarse ahora. Para ello se debe hacer un esquema de las posibles opciones a nivel muy general y discutirlos con los usuarios, analistas de operación y telecomunicaciones definiendo entre todos la mejor estrategia a seguir.

Asimismo deberá de revisar la lista de requerimientos con objeto de comprobar si las opciones posibles cumplen dichos requerimientos. Por último deberá seleccionar la mejor opción en base a un análisis costo-beneficio.

Es indispensable que recuerde que es conveniente utilizar la misma herramienta CASE que ha estado manejando durante el transcurso de este desarrollo de sistema u otra que se pueda integrar a las anteriores para poder generar el código de los programas automáticamente y codificar ciertos detalles realizando interfases con otros lenguajes. Considere en sus soluciones el software CASE generador de código y otros lenguajes de tercera o cuarta generación si se requieren.

- Realizar el diseño físico.

- Actualización del plan.

Quizás se requiera de actualizar o cambiar el plan general, tal que se requiera de más recursos, como resultado del análisis.

- Realizar la reevaluación de costos.

Si se requiere de más recursos para la realización de este proyecto realice nuevamente estimaciones de costos, no olvide utilizar el método de COCOMO.

- Reevaluar la justificación.

Dado los nuevos costos que no se habían tomado en cuenta será necesario actualizar el documento de análisis costo-beneficio empleando nuevamente técnicas de comparación de costos.

- Obtener la aceptación del usuario o del cliente y del responsable de sistemas. Deberá analizarse el nuevo análisis costo-beneficio. El usuario determinará si se continúa o no con el proyecto, determinado si es rentable o no el proyecto y si los objetivos, costos y beneficios no han cambiado. Deberá ser formalizado a través de un documento.

Productos terminados:

Esta documentación será generada automáticamente por la herramienta CASE, a través de su módulo de reportes o accedando cada módulo de la herramienta para imprimir así cada producto generado por el módulo. Con excepción del análisis costo-beneficio y el documento de aceptación.

- Diseño lógico de programas (diagramas estructurados) estructuras de información.
- Casos prueba.
- Diseño físico.
- Plan general actualizado.
- Análisis costo-beneficio actualizado.
- Documento final que indique la aceptación de usuario y del responsable de sistemas.

Involucrados:

- Líder del proyecto.
- Analista de sistemas, operación y telecomunicaciones.
- Diseñador.
- Usuarios.
- Auditor.

Responsables:

- Líder del proyecto.
- Analista.

Técnicas y herramientas a utilizar:

- Top-down.
- Diseño estructurado.
- Herramienta CASE (Categoría Middle-CASE, Lower-CASE) soportando el diseño estructurado para la construcción de los diagramas estructurados.
- CASE soportando la administración de proyectos.
- PERT-CPM o diagrama de flechas.
- GANTT.

- Técnicas de comparación de costos.
- Documentación.

Causas que ocasionan un diseño deficiente:

- El análisis que se tiene es deficiente o no se cuenta con documentación.
- Herramientas no existen o no se utilizan por que no se tiene conocimiento sobre ellas o no son eficientes.
- Las técnicas que se pueden utilizar en esta etapa son desconocidas.
- Cambio de técnica con relación a la utilizada en el análisis.
- Diseño deficiente, de programas es incompleto y no cumple con las especificaciones establecidas.
- El diseño de pruebas no se realiza.

Cursos de acción:

- Puede concertarse una cita para aclarar dudas. Esto no deberá de suceder si usted desde el principio del desarrollo se utiliza una herramienta como CASE y además si las actividades tuvieron la calidad esperada.
- Hacer uso de una herramienta CASE, ya que al hacer los diagramas estructurados tendrán que realizarse:
 - + Cambios continuamente.
 - + Realizar los diagramas estructurados descomponiendo en módulos a diferentes niveles.
 - + Verificar la consistencia.

de no ser así, resultará más costoso hacer correcciones en una etapa más avanzada como lo serían las pruebas.

- La capacitación continua evitará que las técnicas y herramientas sean desconocidas.
- Deberá utilizarse la misma técnica empleada en el análisis para que éste sea congruente con el diseño.
- Utilizar una herramienta CASE para validar y verificar el diseño estructurado.
- Se deberán diseñar los casos prueba asegurando que existe una correspondencia entre la aplicación y los requerimientos. Diseñar casos prueba que cubran todas las alternativas posibles; este diseño de prueba reduce costos. CASE sólo podrá generar datos prueba si tiene ya la estructura del programa.

f) Construcción y pruebas por unidad:

En esta etapa de desarrollo de sistemas, es necesario llevar a cabo la traducción de las especificaciones del diseño a código fuente, y efectuar las pruebas a cada uno de los programas generados y que conforman el sistema.

Construcción.

Tomando en consideración la herramienta ya seleccionada y con la cual se va a construir el sistema no olvide utilizar si éstos son lenguajes de tercera o cuarta generación técnicas de programación que sean consistentes, con las técnicas manejadas en el análisis y diseño como:

Estructurada: Permite realizar programas más fáciles de escribir, verificar, leer y mantener utilizando un número limitado de estructuras de control como lo son: secuencial, selectiva y repetitiva. Además incorpora tanto recursos abstractos y diseño top-down .

Modular: A través de este tipo de programación es posible ser más productivos logrando dividir en programa en módulos, realizando cada uno de ellos una actividad o tarea específica, y donde cada uno de ellos es analizado, codificado y probado por separado.

Top-down: El programa a codificar se realiza descomponiendo el problema a resolver en niveles, tal que se considere cada estructura desde dos puntos de vista, ¿lo que hace?, y ¿cómo lo hace?.

Al programar en lenguajes de tercera o cuarta generación tendrá que:

- + Tomar como base el diseño estructurado creando programas para cada uno de los módulos que ahí se proponen.
- + Al codificar no olvide manejar estándares y de tener un buen estilo de programación.
- + Sus programas deberán ser :
 - * Legibles :Fáciles de leer y comprender.
 - * Portables: Fáciles de instalar en cualquier plataforma.
 - * Modificables: Para realizar los cambios necesarios.
 - * Eficientes: Aprovechar al máximo los recursos de la computadora minimizando la memoria utilizada y el tiempo de proceso o ejecución.
 - * Modulares: Dividirlos en bloques donde cada uno de ellos realice un aparte del conjunto del trabajo.
 - * Estructurados: Cumpliendo reglas de la programación estructurada para facilitar su verificación y depuración.

Si ha tomado la decisión de utilizar el generador de código de la herramienta CASE, el código generado automáticamente y consistente con el diseño estará ya libre de errores, lo que no sucederá si programa utilizando lenguajes procedurales (3GL) o no procedurales (4GL).

Una vez generado el código, puede editarlo utilizando cualquier editor de líneas, en este caso el de la herramienta, por si se desean agregar más procesos.

Al terminar la edición podrá compilar el código y después se podrá generar la aplicación tal que el código compilado se ligue a las librerías adecuadas para crear una aplicación ejecutable libre de errores. Asimismo esta aplicación podrá ser instalada en el medio ambiente que desee.

Debe considerar independientemente de si utiliza una herramienta CASE o un lenguaje procedural, la reutilización de aquellas subrutinas o librerías que han sido almacenadas para hacer referencia a ellas, estandarizadas para que sean fáciles de aplicar y validadas para su fácil integración con el resto del programa. Si reutilizamos esta librerías o código lograremos beneficios como:

- Velocidad.
- Disminución de costos.
- Calidad(Ya que esas librerías están no sólo probadas sino quizás hasta mejoradas).
- Reducción de mantenimiento.
- Documentación debido a que los componentes reusables están bien documentados.

Por último no olvide realizar el plan detallado de pruebas por unidad registrándolo en un administrador de proyectos, considerando las fechas para prueba del plan general, asimismo las actividades que vaya realizando deberá ir las registrando, llevando acabo comparaciones de lo real contra lo planeado realizando ajustes para que se logre terminar el proyecto dentro del tiempo y presupuesto acordado.

Pruebas por unidad.

Las pruebas son realmente importantes ya que de esta manera valoraremos la calidad, determinando el apego de los programas a las especificaciones ya definidas, estas deberán realizarse si sus programas han sido construidos por un lenguaje procedural ya que al ser generados por una herramienta CASE podrán probarse y ya no como pruebas por unidad sino como una prueba del sistema ya que como se cuenta con toda la aplicación solo tendríamos que comprobar su calidad realizando las pruebas con casos previamente diseñados.

Si sus programas están en un lenguaje procedural, considerar lo siguiente:

- Tomar en cuenta los casos prueba que se generaron en la etapa de diseño o utilice herramientas CASE que generen los datos pruebas basándose en sus programas o en el dominio de los datos de entrada al programa, si no las usa capture sus datos prueba y los resultados generando dos archivos respectivamente.

- Realizar primeramente las pruebas para investigar las propiedades estructurales del código fuente.

Utilice una herramienta CASE del tipo analizador estático para detectar errores estructurales como variables no inicializadas, errores entre parámetros formales y reales e incumplimiento de estándares. Esta prueba deberá de repetirse hasta que no existan errores .

- Realizar las siguientes pruebas de caja negra haciendo referencia a la base de datos de prueba y de resultados ya generadas para poder investigar el comportamiento del código fuente sobre los datos prueba.

- + Funcionales: Conocidos los datos de entrada y los resultados esperados verificar los resultados obtenidos. Siendo estas pruebas apoyadas por las herramientas CASE catalogadas como verificadores de prueba o herramientas de prueba dinámica.

- + **Desempeño:** Determinar el tiempo de ejecución, la eficiencia del programa, tiempo de respuesta y uso de dispositivos. Utilice herramientas de prueba CASE llamadas analizadores de tiempo y de optimización.
- + **Tensión:** Tratar de romper intencionalmente la unidad o módulo.

Por último en lo que respecta a pruebas por unidad realice las pruebas de caja blanca como lo es la de:

- + **Estructura:** En base a los casos prueba decidir qué rutas dentro del módulo se verificarán. Utilice las herramientas CASE verificadores de prueba.

Al terminar de realizar estas pruebas estaremos verificando el cumplimiento del factor funcionalidad en cada programa.

Considere además de las herramientas como CASE, utilizar:

- + **Comparadores de archivo** para verificar los resultados esperados contra los resultados obtenidos, tal que se reporten las diferencias.
- + **Rastreadores de programa** los cuales proporcionan información del comportamiento de todos los datos en todo el programa o módulo.
- + **Code auditor:** verifica las estructuras y el código.
- + **Test auditor:** Determina el número de saltos o transferencias realizados y no realizados durante la prueba.
- + **Generador de diagrama de flujo:** Genera un diagrama de flujo del código actual , a través del cual se muestra dónde los datos son utilizados para su posterior análisis

Recordar que se tratará de encontrar errores no de ocultarlos, hay que tener la mente destructiva si se requiere el éxito de las pruebas.

Por último deberá documentar sus programas siguiendo estándares y realizar el plan detallado de pruebas del sistema sin olvidar de utilizar su herramienta para la administración del proyecto, registre aquellas actividades que ya se cumplieron, realizando una comparación contra el plan general para llevar a cabo acciones ajustes para que se logre terminar el presupuesto dentro del tiempo y presupuesto acordado.

Productos terminados:

- Listado de programas documentados.
- Base de datos prueba.
- Resultados de las pruebas y de resultados esperados
- Programas validados, depurados y verificados.

- Reporte que muestre la diferencia entre los resultados de las pruebas para cada programa.
- Plan de pruebas detallado.
- Plan actualizado, por los ajustes sin afectar el tiempo y el presupuesto definido para el proyecto.

Involucrados:

- Líder del proyecto.
- Analista.
- Diseñador.
- Programador.
- Auditor.

Responsables:

- Líder del proyecto.
- Programador.

Técnicas y herramientas:

- CASE generadores de datos prueba.
- Editores de línea.
- Librerías o biblioteca de rutinas.
- Generadores de código (Categoría Lower-CASE).
- Herramientas CASE (Analizadores de tiempo, optimizadores y verificadores).
- Lenguajes de cuarta generación.
- Técnicas de programación (Top-down, estructurada, modular).
- Lenguajes de programación procedurales .
- Analizadores estáticos de programas.
- Paquetes de rastreo de programas.
- Pruebas de caja negra.
- Pruebas de caja blanca.
- Herramientas Code-Auditor, Test-Auditor.

- Generador de diagramas de flujo del programa.
- CASE soportando la administración de proyectos.
- PERT-CPM o diagrama de flechas, GANTT.
- Software para generar archivo de datos prueba y resultados.
- Documentación.

Causas que ocasionan una construcción y pruebas por unidad deficiente:

- Mala interpretación de las especificaciones del análisis y diseño.
- Estilo y lenguaje de programación complejo, dada la falta de estándares de programación, lo cual implica que no se logre un código de calidad uniforme entre todos los programadores.
- Se pierde mucho tiempo en traducir el diseño a código ya que no es semejante.
- Código no portable.
- No existe documentación interna en los programas.
- No se realizan las pruebas por unidad.
- Al realizar las pruebas no se cuenta con los casos ni con los datos prueba.
- No se cuenta con los manuales del sistema operativo ni del lenguaje de programación o de las herramientas seleccionadas como CASE.
- No se utilizan las técnicas y herramientas.
- Las especificaciones son inadecuadas.
- No existe validación a los datos de entrada.
- No se utilizan técnicas de programación.
- Fallas a leer o grabar información, acceso incorrecto a los elementos de un arreglo, variables no definidas, ciclos infinitos, abuso de variables, condiciones invertidas e incompletas, anidamientos inadecuados, incorrecto paso de parámetros.

Cursos de acción:

- Existe mala interpretación de las especificaciones porque las dudas que se tienen no son aclaradas. Por lo tanto los programadores, deberán aclarar sus dudas con el analista o diseñador, si no se llevará acabo una mala traducción del diseño al codificar.
- Si sólo se dispone de un lenguaje de programación no hay nada que hacer. Sin embargo para que su lenguaje no sea complejo debe considerarse al elegirlo:
 - + Area de aplicación.
 - + Complejidad algorítmica.
 - + Entorno en el que se ejecutará el lenguaje.
 - + Consideraciones de rendimiento.
 - + Complejidad de las estructuras de datos.
 - + Disponibilidad de un compilador.

En lo que respecta al buen estilo de programación, éste inclusive puede superar las deficiencias de un lenguaje de programación primitivo; si su estilo es pobre de nada servirá el excelente lenguaje que tenga.

Su estilo de programación será bueno si el código es fácil de comprender, si es sencillo y elegante.

Las acciones a evitar para poder tener un buen estilo de programación serán:

- + No debe ser demasiado complicado.
- + Si maneja instrucciones condicionales evitar then o then if o else nulos.
- + No realizar muchos anidamientos ni en ciclos ni en instrucciones condicionales.
- + No suboptimizar.
- + No emplear identificadores para propósitos múltiples.
- + Utilizar librerías ya existentes.

Además haga uso de los estándares definidos por el departamento o que están definidos por la metodología seleccionada:

- + Instrucciones.
- + Indentación si se requiere.
- + Reglas asociadas con los módulos a programar.
- + Nombres de datos.
- + Nombres de módulos.
- + Nombres de archivos.
- + Nombres de registros.
- + Etc.

Son indispensables los estándares para que el estilo de codificación sea similar es por eso que se deben definir y documentarse para uso de los programadores. Si se utilizan herramientas automatizadas será fácil verificar el uso de los estándares, además al no utilizarlos será difícil entender los programas desarrollados por otras personas.

- Al elegir un lenguaje seleccionar el que más se adecúe a su diseño o a la técnica de diseño elegido, ya que si no, dedicará mucho tiempo en la traducción.

- Si la aplicación a desarrollar tendrá que ser transportada a otro procesador, compilador o trabajar con una nueva versión de sistema operativo o inclusive ser utilizado por otros paquetes, será la portabilidad un requisito indispensable.
- El que no exista documentación interna causará problemas al realizar un mantenimiento al programa, es por eso que al terminar el programa debe incluirse la siguiente información por programa y por cada módulo que lo conforma:

Por programa:

- + Nombre del programa.
- + Tarea que realiza.
- + Nombre del programador.
- + Fecha en que se creó el programa.
- + Número de versión o actualización.
- + Fecha de la última actualización.
- + Lista de módulos del programa indicando quién llama a quién.
- + Conexiones con otros programas.
- + Entradas y salidas.

Para cada módulo definir:

- + Su función.
- + Descripción de variables.
- + Autor del módulo.
- + Fecha de la última modificación.

Además por cada programa no se debe olvidar utilizar:

Comentarios descriptivos en el programa para:

- + Describir los bloques del código.
- + Dejar líneas en blanco y sangrías para que el código sea entendible y legible.

- Las pruebas de cada módulo y programa deberán llevarse al terminar de codificar para lo cual deberán de tomarse en cuenta los casos prueba previamente diseñados así como los datos de prueba también ya creados, de modo que se eliminen todos los errores si la prueba resulta ser exitosa.
- Se deben tener en el área de trabajo los manuales necesarios para trabajar y sobre todo tenerlos actualizados y en buen estado.
- Muchas veces los programadores trabajan a mano, hay que tomar en consideración los editores tan poderosos que existen actualmente, sobre todo los que acompañan al lenguaje de programación. El hecho de utilizar los editores es por que a veces manejan comandos que reducen el tiempo que requiere el programador para editar su programa, las herramientas que rastrean el programa o los analizadores estáticos le ayudarán rápidamente a encontrar sus errores.

- Para que las especificaciones no sean inadecuadas deberá utilizar lo propuesto en la fase de diseño.
- Muchas veces no se utilizan técnicas de programación como la estructurada, modular o la orientada a objetos, porque no se conocen o porque no se sabe cómo aplicarlas, las técnicas de programación deberán ir acorde a lo definido en el diseño. En un dado caso que la técnica no se domine, deberá de capacitarse.
- Deberán construirse los procesos que permitan validar cualquier información que se introduzca al sistema, más aún si el lenguaje no realiza por el tipo de datos que maneje, una validación automática.
- El abuso de variables, ciclos infinitos, exceso de anidamientos etc., es causa de la falta de experiencia en el lenguaje, y en programación puede evitarse esta problemática si utiliza lenguajes de cuarta generación o herramientas CASE.

g) Pruebas del sistema:

En esta fase se realizarán acabo tanto la prueba del sistema, como la prueba de aceptación por parte del usuario, asegurando así que el sistema cumpla con los requisitos y rendimiento solicitado por el usuario.

Actividades:

- Generar la base de datos prueba utilizando una herramienta CASE o tomando como referencia los casos prueba diseñados anteriormente tal que capture los datos y cree el archivo tanto de datos como de resultados.
- Realice la prueba de integración .
Para integrar todos los módulos o programas, verificar la interfase entre ellos y asegurar que cada módulo así como los que se vayan integrando cumplan con los requerimientos y especificaciones del diseño. Utilice para ello los siguientes enfoques:
 - + Top-down: Se empieza probando el módulo raíz y posteriormente todos los demás módulos subordinados se van llamando y probando hasta que todos los módulos sean incorporados.
 - + Bottom-up: Cada uno de los módulos del nivel más bajo se prueba individualmente, después los siguientes módulos son llamados por los ya probados hasta que todos los módulos se han incluido en la prueba.
 - + Big-bang: Ya probados todos los módulos individualmente todos se encadenan y se realiza la prueba.
 - + Sandwich: Combina tanto el enfoque top-down como el bottom-up. Los niveles de arriba se combinan con los del centro y los de abajo también. Utilizando el enfoque to-down para el nivel superior y el bottom-up para el inferior.

Haga uso de herramientas de comparación, CASE verificador y Drivers que simulan la llamada a módulos e inclusive las ejecutan. Este tipo de herramientas agilizarán el proceso de esta etapa.

- Probar el sistema realizando:

- + Pruebas de recuperación: Hacer fallar al sistema verificando que la recuperación sea apropiada siendo ésta quizá en forma automática o mediante la intervención de una persona. Evaluar el tiempo de recuperación.
- + Pruebas de seguridad: Verificar si el sistema consta de mecanismos de protección, tratar de penetrar al sistema y determinar si el costo de penetración es mayor que el valor de la información obtenida mediante la penetración.

También será importante verificar la seguridad que el sistema proporcione, identificando así los riesgos de éste al cometer algunos errores en los datos de entrada, para ello será importante utilizar técnicas como:

Análisis de árbol de fallos: Mediante éste será posible determinar las consecuencias de una secuencia de fallos.

Lógica de tiempo real: Mediante un modelo sucesión se prueba la seguridad de los componentes del sistema.

Redes de Petri: Determinan los riesgos más p

- + Pruebas de resistencia: A través de esta prueba se enfrentará al sistema a situaciones anormales, tal que el sistema al ejecutarse demandará muchos recursos.
- + Pruebas de rendimiento: Asegurar el rendimiento deseado del sistema, rendimiento que sea probado desde la prueba por unidad.

- Realizar el plan detallado de la prueba de aceptación el cual deberá ser revisado y aceptado por el usuario.
- Generar la base de datos prueba para llevar a cabo la capacitación y la prueba de aceptación del sistema.
- Plan de capacitación detallado a algunos usuarios para prueba del sistema.
- Preparación del material de capacitación (hardware, software, manual del usuario y los documentos que firmará el usuario si es aceptado el sistema).
- Capacitación a sólo algunos de los usuarios siendo posibles entrenadores.
- Realizar la prueba de aceptación para determinar si el sistema satisface los requerimientos del usuario, validando la facilidad de uso del sistema .

Puede llevar a cabo:

- + Pruebas alpha: aquí el sistema es probado por un sólo usuario en el lugar de desarrollo. Ahí el líder del proyecto tomará notas de los problemas o errores en el uso del sistema.
- + Pruebas beta: el sistema es probado en diferentes lugares por diferentes usuario, estos registrarán los problemas y errores y se lo harán llegar al líder del proyecto.

- Realizar mínimas correcciones en caso de ser necesarias.

- Los miembros del equipo de desarrollo que intervengan en ésta etapa deberán registrar aquellas actividades que van terminando, el líder del proyecto realizará pequeños ajustes como resultado de las desviaciones al comparar lo real contra lo planeado. Esto no deberá afectar el tiempo y presupuesto del proyecto.

Productos terminados:

- Manual del usuario terminado.
- Manual técnico.
- Resultados de las pruebas.
- Base de datos prueba y de resultados esperados.
- Reporte que muestre las diferencias entre los resultados esperados y reales del sistema.
- Sistema depurado, validado y verificado.
- Documento formal que certifique la aceptación del sistema por parte del usuario.
- Plan general actualizado, con mínimos ajustes para que no se afecte el tiempo y presupuesto definido para el proyecto..
- Plan detallado de pruebas de aceptación.
- Plan detallado de capacitación.

Involucrados:

- Responsable de sistemas.
- Personas que realicen pruebas de integración y del sistema(audidores), en este tipo de pruebas se debe excluir a los desarrolladores del proyecto.
- Usuarios.
- Programadores.
- Auditores.
- Líder del proyecto.

Responsables:

- Líder del proyecto.
- Programadores.

Técnicas y herramientas:

- Generadores de datos prueba.
- Software administrador de proyectos.

- Herramientas CASE del tipo verificador y generadores de datos prueba.
- Pruebas alfa y beta.
- CASE soportando la administración de proyectos.
- PERT-CPM o diagrama de flechas y GANTT.
- Documentación.

Causas que ocasionan deficiencia en las pruebas:

- No existen casos pruebas o son inadecuados, si existen, muchas veces no se generan los datos prueba de acuerdo a los casos.
- No se detectan errores o existen problemas para detectarlos.
- No existe una planeación de las pruebas.
- No se realizan las pruebas.
- No se conocen las técnicas y herramientas para llevar a cabo las pruebas.
- Material de capacitación no existe.
- Capacitación deficiente.
- Manuales confusos e incompletos.
- Las personas que capacitan no tienen las cualidades necesarias para capacitar y convencer a los usuarios de los beneficios del sistema.

Cursos de acción:

- Es recomendable que el diseño de los casos pruebas se realice en la fase de diseño y que cubran todas las posibles alternativas dentro del sistema; los datos deben de generarse en base a estos casos.
- Si no se tienen casos prueba así como las herramientas para hacer las pruebas puede no detectarse ningún error.
- Deberá realizarse la planeación de las pruebas ya que si no, éstas no se realizarán por empalme de actividades o por que ya no hay tiempo para llevarlas a cabo. Debe indicar en este plan:
 - + Objetivo de las pruebas.
 - + Cómo las pruebas serán administradas y que criterios serán utilizados para juzgar cuando la prueba está completa.
 - + Deberán indicarse las herramientas y las técnicas a utilizar con esta prueba.
- Se deberán realizar las pruebas al sistema si se quiere evitar que los errores aparezcan durante su operación, logrando la insatisfacción del usuario.

- Vuelve el problema de la falta de investigación en nuevas tecnologías así como la capacitación en ellas.
- El material para capacitación puede ser un manual, material para clase, un video e inclusive un sistema con iconos o menús pop-up que explique el funcionamiento del sistema, es muy importante tomar en consideración el nivel educativo de las personas que capacitará.
- Deberá de preparar datos para el usuario para visualizar el comportamiento de éstos a través del sistema.
- Los manuales del usuario pueden empezarse a elaborar desde que el prototipo es aceptado. Si estos están incompletos pueden ocasionar que no se despejen las dudas o que sean confusos . Por lo tanto se recomienda que contengan por lo menos la siguiente información:

+ Introducción.

Funciones que ejecuta.
Desarrolladores del sistema.
Nombres de las personas que pueden atender sus dudas.
Requerimientos de hardware y software.
Limitaciones del sistema.

+ Principios y procedimientos.

¿Cuáles son las salidas? y ¿Cómo deben ser interpretadas?
¿Cuáles son las entradas? y ¿Como se obtienen?
¿Cómo se opera el sistema? y el tipo de menú e interfases que maneja.

Procesos generales.
Formatear discos.
Respaldar información.
Entrada y salida del sistema.

+ Tutorial : mostrar paso a paso el funcionamiento del sistema con un ejemplo usar narraciones y mostrar las pantallas, así como los datos de entrada y salida.

+ Referencia.

Para cada función :

Describirla.
Indicar al estructura de comandos.
Indicar cuándo y como utilizarla.
Mostrar que errores pueden ocurrir y qué hacer en caso de que ocurran.
Ejemplos. '

Por error.

¿Cómo se identifican?
¿Qué significan?
¿Qué se puede hacer?

+ Como instalar el sistema antes de su primer uso.

- + Indicarle al usuario como crear sus propias aplicaciones (para generar la información deseada, hacer consultas a la base de datos mediante *Queries*).
- No debe ser muy técnico, ni muy teórico en la capacitación, debe hacerse la presentación lo mas atractivo que pueda ser utilizando un lenguaje accesible y detallado.

h) Implantación:

En esta fase se terminarán de capacitar al resto de los usuarios, se instalará y probará el equipo, se convertirán o se construirán los archivos de datos verificando que el sistema quede instalado en la empresa. Esta etapa deberá estar apoyada por los vendedores, o alguien externo y el staff de desarrollo ya que es necesario que sea realizada por el departamento de operación liberando así de estas actividades al personal de desarrollo.

Actividades:

- Capacitar al implementador del sistema. Este deberá conocer los aspectos generales del sistema, lugares donde será instalado, sus usuarios, los beneficios que con este sistema se logran así como el hardware y software que utilizará.
- Evaluar los conocimientos adquiridos del implementador.
- Verificar que el sistema sea de calidad y haya sido aceptado por el o los usuarios. Esta actividad deberá omitirse para el sistema desarrollado con la guía anterior ya que si no se estaría incurriendo en una inspección final innecesaria. Lo que si deberá de revisar será el documento que certifique la aceptación del sistema por parte del usuario.

Más sin embargo si su sistema fue comprado puede realizar una evaluación del sistema determinado su cuenta con los factores de confiabilidad, facilidad de uso, funcionalidad, realizabilidad, seguridad e interoperabilidad los cuales serán fácilmente visibles al ejecutar el sistema. Quien más que un usuario para evaluarlo asignando una puntuación de 0 a 5 a cada uno de estos factores, sumando los puntos y promediándolos determinando así si el sistema es:

CALIDAD	PUNTOS
<i>Excelente</i>	5
<i>Muy bueno</i>	≥ 4 y < 5
<i>Bueno</i>	≥ 3 y < 4
<i>Regular</i>	≥ 2 y < 3
<i>Pésimo</i>	≥ 0 y < 2

Si el factor de funcionalidad no llegara a cumplirse su sistema no deberá considerarse con calidad.

- Realizar el plan detallado de instalación, revisarlos con los usuarios y el responsable de sistemas de tal manera que se obtenga su autorización.
- Preparar el ambiente técnico para la instalación. Asegurar que el mobiliario y el equipo de cómputo está instalado y probado correctamente.
- Preparar si se requiere el material para capacitación de capturistas , dar aviso formal al período de captura.
- Capacitar a los capturistas, si se requiere.

- Conversión de datos o preparar los datos y capturarlos.
- Plan de capacitación al resto de los usuarios.
- Tener preparado el material de capacitación(equipo de cómputo, manual de usuario, material para exposición) así como los datos con los que se trabajará.
- Capacitar al resto de los usuarios.
- Instalación del sistema.
- Registrar aquellas actividades que fueron realizadas para posteriormente compararse contra el tiempo que se había planeado para ellas y realizar pequeños ajustes.

Para llevar a cabo esta instalación se podrá seguir cualquiera de las siguientes alternativas:

- Cortar totalmente el sistema antiguo y empezar con el nuevo. La instalación podrá efectuarse fuera de horarios de oficina.
- Operar un paralelo desfasado para dar confianza, fijando la fecha de inicio.

Involucrados:

- Responsable de sistemas.
- Líder del proyecto.
- Analistas.
- Diseñadores.
- Programadores.
- Usuarios.
- Consultor.
- Vendedores.
- Operación.

Responsables:

- Líder del proyecto.
- Operación.

Técnicas y herramientas:

- Utilerías para conversión de archivos.
- CASE soportando la administración de proyectos.
- Software para la administración de proyectos.

Productos terminados:

- *Plan de implantación detallado.*
- Plan de capacitación.
- Material de capacitación.
- Nuevas bases de datos o archivos.
- *Plan actualizado, por mínimos ajustes sin llegar a que afecte el tiempo y presupuesto total definido para el proyecto..*

Causas que ocasionan la deficiencia en la instalación.

- *El sistema no tiene calidad por consecuencia de un mal análisis, diseño y programación, por consecuencia el sistema no debe haber sido aceptado por el usuario.*
- Material de capacitación no existe .
- Capacitación deficiente.
- No se cuenta con el manual técnico actualizado.
- Manuales confusos e incompletos.
- *Las personas que capacitan a los usuarios no tienen las cualidades y habilidades necesarias para capacitar y convencer a los usuarios de los beneficios del sistema:*
- No se prepara el ambiente técnico para la instalación.
- No se conoce donde instalar el sistema.

Cursos de acción:

- *La persona que realizará la instalación del sistema deberá asegurarse de que el sistema tenga la calidad para lo cual deberá conocer totalmente el sistema. No deberá instalarse si este no ha sido aceptado por el usuario, para lo cual deberá verificarse la existencia de un documento formal que certifique su aceptación.*
- *Deberá preparar con tiempo el material para la capacitación si no se cuenta con el material de la fase anterior, este material puede ser un manual, material para clase, un video e inclusive un sistema con iconos o menús pop-up que explique el funcionamiento del sistema. Es muy importante tomar en consideración el nivel educativo de las personas que capacitará.*

- No se debe ser muy técnico, ni muy teórico en la capacitación, debe hacerse la presentación lo más atractiva que pueda utilizando un lenguaje accesible y detallado.
- Deberán prepararse datos para que el usuario pueda visualizar el comportamiento de éstos a través del sistema.
- A lo largo del desarrollo del sistema cada una de las personas del equipo de desarrollo deberá ir generando su documentación actualizándola por los cambios que puedan ocurrir, tal suerte que al final se tenga un manual completo y actualizado para futuros mantenimientos.
- Deberá de prepararse el ambiente técnico para instalar con anticipación el hardware y quizás una nueva versión del software de que será utilizado como plataforma, asimismo debe de considerarse el tiempo planeado de llegada del equipo y del software, el tiempo que se requerirá para su instalación para que pueda instalarse el nuevo sistema en la fecha convenida.
- Es indispensable conocer físicamente donde se llevará a cabo la instalación del sistema dentro de la empresa, para lo cual se tendrá que leer la información generada en el análisis.

i) Proporcionando garantía.

Después de que el sistema ha pasado de la etapa de producción a la de operación ya no evaluaremos si el nuevo cumple con los requerimientos, ya de antemano se sabe que éste ya no tiene errores.

Sin embargo el departamento de sistemas establecerá un período de garantía de 2 a 6 meses teniendo opción a cualquiera de las siguientes alternativas:

- El líder del proyecto podrá permanecer en el lugar donde reside el sistema en operación.
- Hacer una llamada a cualquier persona que ha estado involucrada en el desarrollo del proyecto para atenderlo en forma inmediata.
- Llamar al departamento de sistemas para reportar el problema tal que este se registre y se haga llegar a la persona más indicada para atenderlo en el menor tiempo posible.

Usted podrá definir si se selecciona una de estas alternativas para todo el período de tiempo o si se realiza una combinación de todas en ese período.

Si surgen nuevos requerimientos o solicitudes de cambios, documéntelos quizás eso forme parte de un nuevo proyecto y no trate de ofrecer un mantenimiento al proyecto viejo.

Involucrados:

- Líder del proyecto.
- Usuario(s).
- Staff de desarrollo.

Responsables:

- Algún miembro del staff de desarrollo y el líder del proyecto.

Conociendo los objetivos de la calidad, los problemas que se presentan en el área de desarrollo de sistemas y contando ya con una guía para ese desarrollo se ha decidido que las siguientes estrategias son las más adecuadas para afrontar los problemas en el área de desarrollo con el fin de ir logrando la calidad en nuestros sistemas.

1. Organización: Establecer el concepto del proceso de calidad con la alta administración (directivo o responsable de sistemas) determinando cuál sería la organización necesaria del área de desarrollo de sistemas para apoyar este proceso.

Al utilizar una herramienta CASE en el desarrollo de sistemas sólo se requerirá de un verdadero líder del proyecto que realmente sea el administrador del equipo siendo el líder y motivador, quién planeará las actividades a llevarse a cabo, manteniendo un control con el fin de lograr un proyecto exitoso.

Se requerirán además analistas con la capacidad de diseñar interfases y de realizar modelaciones tanto de datos como de procesos, así como analistas-programadores con la capacidad de diseñar y construir programas sobre todo cuando estos sean desarrollados utilizando lenguajes 3GL o 4GL.

2. Formación: Crear en todo el personal una nueva cultura tanto de calidad como computacional o de informática (en metodologías, técnicas y herramientas) basado el primero en programas de información al personal apoyado por la alta dirección y a través de una motivación hacia un deseo común. En cuanto a la cultura de informática esta se logra a través de un programa continuo de capacitación mejorando y actualizando los conocimientos y habilidades de las personas, previniendo riesgos e incrementando la calidad y productividad. Ese programa de capacitación será elaborado conforme a un diagnóstico de necesidades considerando la situación real y la situación ideal a la que se quiere llegar, llevando a cabo un análisis comparativo (relación de perfiles contra puestos). Será necesario también que el programa sea realizado considerando no sólo las necesidades del personal de sistemas sino sus deseos.

El nivel y escolaridad que se requiere que posean las personas involucradas en el desarrollo e implantación del sistema es de licenciatura o superior, en el área de sistemas computacionales requiriendo además de amplia experiencia y conocimiento del negocio.

Por ejemplo un Licenciado en Informática el cual puede evaluar, seleccionar e implantar un sistema computacional . Un Licenciado en Sistemas Computacionales es aquel capaz de diseñar, desarrollar e implantar los sistemas computacionales para administrar la información necesaria para la toma de decisiones , además de diseñar métodos y procedimientos que contribuyan a optimizar los recursos de la empresa. Y el Ingeniero en Sistemas Computacionales podrá desarrollar aquellos sistemas más complejos como sistemas operativos, manejadores de bases de datos tal que estos sean no sólo considerados como un software de aplicación sino como una nueva tecnología. El conocimiento de varios lenguajes de programación y el desarrollo de nuevas metodologías lo caracterizan.

3. Comunicación: Deberá existir una comunicación permanente entre los miembros del equipo de desarrollo de sistemas, para tener presente el avance que se ha logrado con la calidad en lo que respecta al proceso de desarrollo de sistemas e implantación.

4. Desarrollo en equipo: Todo proyecto que involucre la realización de un sistema de información tomando en consideración su magnitud, deberá ser realizado por un equipo de personas altamente capacitadas y de calidad.

5. Monitoreo: Dar seguimiento en lo que respecta a la ejecución del proyecto, detectando desviaciones conforme a lo planeado, analizando sus causas y corrigiendo las desviaciones con previa autorización.

6. El usuario del sistema: Deberá de ser la prioridad absoluta para el departamento de sistemas. La satisfacción se logrará a través de la calidad del sistema desarrollado.

7. Participación de los proveedores de información: Serán importantes contribuidores al desarrollo de nuevos sistemas con calidad. El nivel de calidad del sistema dependerá de la calidad de los requerimientos obtenidos. Si se desea desarrollar un sistema de información en menos tiempo y altamente confiable, se requerirá de su alta participación .

8. Proceso: La calidad deberá asegurarse a través del control del proceso del desarrollo de sistemas aplicando la técnica adecuada como la prevención. Una prevención que implicará hacer las cosas bien a la primera, además de anticiparse a los problemas conociéndolos al igual que sus causas, si llegara a suceder un problema no contemplado se deberán buscar sus causas eliminándolas lo más pronto posible desde su origen. No se tratará de lograr la calidad por inspecciones sino de fabricarla con el mínimo costo donde cada persona haga bien su trabajo y lo autocontrole.

9. Nunca parar: Se deberá buscar una mejora continua en los procesos, tiempo, gente y la tecnología.

- Proceso: Mejorar el proceso a través del uso de nuevas tecnologías de información.
- Tiempo: Reducción en tiempo de desarrollo y en costos.
- Hombre: Formación de grupos y equipos de trabajo.
- Tecnología: No copiar ideas de sistemas ya existentes: tratar de simplificar de transformarlos.

No se trata de que la mejora sea súbitamente sino al contrario deberá ser en forma gradual, para lo cual no se tratará solamente de introducir una innovación (CASE) ó una nueva metodología de desarrollo de sistemas, realmente se requerirá de muchos esfuerzos reconocidos ante todo, por el personal involucrado en el proceso de desarrollo del sistema.

10. Los nuevos sistemas: Si la calidad como se definió en un principio de esta tesis no es sólo satisfacer al cliente sino sobrepasar sus expectativas, debemos a través de la mejora continua lograr este objetivo desarrollando cada vez en menos tiempo y con más altos niveles de calidad y confiabilidad, siendo el elemento clave en esto la tecnología de información.

11. Promoción de la calidad: Dar a conocer continuamente a todo el personal la filosofía de calidad, los objetivos, la misión de la empresa, el plan y las políticas de calidad, todo esto para que se de un cambio de mentalidad.

- Retroalimentar los resultados a través de una publicación sobre calidad.
- Presentar los mejores sistemas en convenciones.
- Dar un reconocimiento a las mejores ideas.
- Llevar a cabo un intercambio de experiencias a través de reuniones(pueden formarse círculos de calidad).

12. Los recursos físicos: Para desarrollar un sistema de información con calidad requerirá invertir en tecnología de información como lo son las herramientas CASE así como en capacitación a todo su personal en metodologías y técnicas antes de aprender a utilizar estas herramientas. Además si desea tener una herramienta CASE poderosa que automatice todo el ciclo de vida del sistema o en dado caso diferentes herramientas modulares, será indispensable contar con un equipo computacional ya sea una estación de trabajo con un procesador 486 o *pentium* de ser posible y con suficiente capacidad en disco y de memoria (dependerá de la herramienta, pero considere que la mayoría consume muchos recursos), o una minicomputadora y de ser posible una herramienta que pueda trabajar en red siendo esto lo mejor, para el equipo de desarrollo de sistemas, ya que podrán compartir toda la información que se encuentre almacenada en el repositorio y podrá avanzarse más rápido al estar desarrollando el sistema.

13. Participación total: Todas las personas serán responsables de que se dé y se mejore la calidad, se requiere de un compromiso y de una participación de todos hacia ese nuevo reto sin fin.

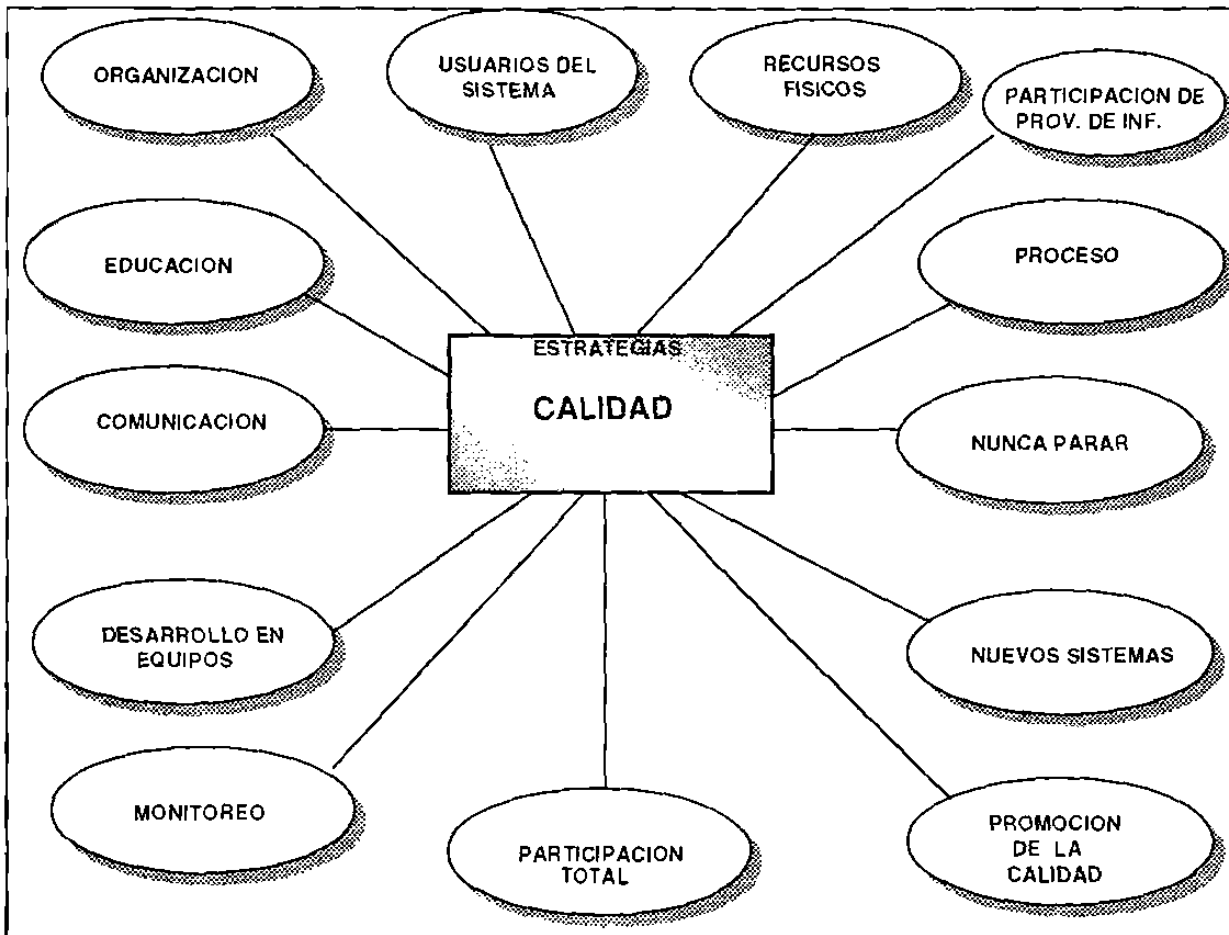


Fig.6.1 Estrategias para el logro de la calidad

CONCLUSIONES

Las respuestas a los diferentes problemas que se han presentado en el área de desarrollo de sistemas serán resueltos si logramos obtener calidad.

Calidad que debe irse dando a lo largo del desarrollo del sistema si se requiere un producto que no sólo logre cumplir los requerimientos de los usuarios sino que sobrepase sus expectativas, además que funcione, cumpla con estándares en un principio definidos, sea fácil de usar y sobre todo se logre terminar respetando el tiempo y presupuesto establecido.

El sistema con calidad será el resultado de llevar a cabo las actividades que conforman la guía propuesta para desarrollar un sistema, donde una combinación de elementos deben ser aplicables a cada actividad. Siendo estos elementos el contar con personas capacitadas en metodologías, técnicas y herramientas que como CASE aseguran el éxito por la consistencia que se logra al utilizarlas. Trabajar en equipo con efectividad, donde cada persona esté comprometida a realizar bien su trabajo y autocontrolarlo, dirigidos por un líder con la capacidad de guiar sin dominar, desarrollando las potencialidades de los miembros de su equipo, motivándolos a trabajar por un objetivo común y proporcionándoles los recursos necesarios para desempeñar con calidad su trabajo, involucrando además a personas de otras áreas (operativas, telecomunicaciones y redes, etc.) así como a consultores y auditores con amplia experiencia. Basarse además en estándares definidos por alguna institución internacional como ISO. Aplicar algún modelo o método para lograr la calidad como sería la prevención, anticipándose a los problemas a través de su conocimiento o a través de la búsqueda de sus causas, eliminándolas lo más próximo posible a su origen dentro del proceso. Y por último administrando todo el proceso del desarrollo de sistemas basado en planes y controles para lograr que el sistema sea entregado dentro del tiempo y presupuesto planeado.

Creo que es importante recalcar que en lo que respecta a la tecnología de información como lo es CASE, ésta deberá utilizarse adecuadamente para lograr los factores de calidad en cada fase del desarrollo de sistemas tanto como soporte a la administración de todo el proyecto, como a la generación de prototipos para definir requerimientos con calidad, indispensable para llevar a cabo tanto la modelación de datos como de procesos y construir el sistema en forma automática libre de errores.

CASE realmente no sólo es una herramienta sino toda una filosofía que involucra diferentes tecnologías de información que permitirá involucrar por fin al usuario durante todo el proceso del desarrollo de sistemas logrando así que éste tenga más responsabilidades en ese desarrollo consiguiéndose así producir sistemas más rápidos y menos costosos, además de que con CASE se logrará un que el proceso de desarrollo sea totalmente disciplinado basado en estándares.

Sin embargo debemos de pensar en mejorar continuamente aquello que hacemos bien fijándonos nuevos objetivos, definiendo nuevas estrategias para alcanzarlos, mejorando el proceso de desarrollo de sistemas y actualizándose en nuevas metodologías, técnicas y herramientas a través de programas de capacitación definidos en común acuerdo entre jefe y subordinados. Recuerde que este mejoramiento implicará desechar aquello que ya no contribuye a lograr niveles más altos de calidad.

Por último quisiera recalcar que el tema de la calidad es altamente apasionante más aún si uno trata de relacionarlo con el departamento de sistemas, considero que existen muchas oportunidades para que éste logre ser un departamento de calidad total apoyado no sólo por la tecnología de información sino por otros métodos, procesos, modelos o filosofías de calidad ya probados en áreas de producción y las cuales pueden llegarse adaptar. Hago referencia a ello porque existen otros métodos como CWQC o TQC, KAIZEN, POKA-YOKE, QFD los cuales podríamos ir adaptando al desarrollo de sistemas para actualizar las ya existentes o crear nuevas metodologías o guías de desarrollo, involucrándose así en un proceso de mejora continua.

Sólo me resta afirmar lo que muchos ya han dicho, la calidad es y será la clave del éxito para ser cada vez mejores y más competitivos.

BIBLIOGRAFIA DE REFERENCIAS

- [AUG,91] August, Judy H. Joint Application Design.
E.U.A., Yourdon Press Computing Series, 1991.
- [BUE,90] Buenrostro, Ernesto. Control Total de Calidad a su Alcance.
México, EDIMSA, 1990.
- [CRO,90] Crosby, Philip B. La Calidad No Cuesta.
México, Cía. Editorial Continental, S.A. de C.V., 1990.
- [CRO,91] Crosby, Philip B. Calidad Sin Lágrimas.
México, Cía. Editorial Continental, S.A. de C.V., 1991.
- [EDW,93] Edwards, Perry. Systems Analysis & Design.
E.U.A., McGraw-Hill, 1993.
- [FEI,91] Feigenbaum, Armand V. Control Total de Calidad.
México, Cía. Editorial Continental, S.A. de C.V., 1991.
- [FIS,88] Fisher, Alan S. CASE Using Software Development Tools.
E.U.A., John Wiley & Sons, Inc., 1988.
- [FRE,87] Freeman, Peter. Software Perspectives. The System is the Message.
E.U.A., Addison-Wesley Publishing Company, 1987.
- [GIB,88] Gibson Michael L. "A Guide To Selecting CASE Tools".
En: Datamation. E.U.A., (1998), pp.65-66.
- [GIB,89] Gibson, Michael Lucas. "Implementing the Promise".
En: Datamation. E.U.A., (1989), pp.65-67.

- [GIT,89] Gitlow, Howard S. y Shelly J. Gitlow. Cómo Mejorar la Calidad y la Productividad con el Método Deming. Colombia, Ed. Norma, 1989.
- [HUG,90] Huges Cary T. y John D. Clark. "The Stages of CASE Usage". En: Datamation. E.U.A., (1990), pp.42-45.
- [ISH,86] Ishikawa, Kaoru. ¿Qué es el Control Total de Calidad? Colombia, Ed. Norma, 1986.
- [JOH,91] Johnson, James R. The Software Factory. Managing Software Development the Maintenance. 2da. edición. E.U.A., Computer Weekly Publications, 1991.
- [JUR,90] Jurán, Joseph M. Jurán y el Liderazgo para la Calidad. Manual para Ejecutivos. Madrid, Ediciones Diaz de Santos, S.A., 1990.
- [MAR,90] Martin, James. Information Engineering. Book I, II, III. E.U.A., Prentice Hall, 1990.
- [MAR,91] Martin, James. Rapid Application Development. New York, McGraw-Hill, 1991.
- [MCC,93] Carma McClure. CASE la Automatización del Software. E.U.A., Addison-Wesley Iberoamericana, 1993.
- [MOL,93] Möller, K. H. y D.J. Paulish. Software Metrics. U.S.A., IEEE Computer Society Press, 1993.
- [PAT,92] Patrinostró, Frank y Steve Ayer. Documenting the Software Development. E.U.A., McGraw-Hill, 1992.

- [PRE,93] Pressman, Roger S. Ingeniería del Software. Un Enfoque Práctico.
3a.edición.
México, McGraw-Hill, 1993.
- [POW,90] Powers, Michael J, Paul H. Cheney, Galen Crow. Structured
System Development.
E.U.A., Boyd & Fraser Publishing Company, 1990.
- [ROY,90] Roy, Ranjit K. A Primer on the Taguchi Method.
E.U.A., Competitive Manufacturing Series, 1990.
- [RUI,93] Ruiz González Carlos. "Reingeniería: ¿Moda o Concepto Crucial?
En: Expansión. México, Diciembre(1993), pp.170-171.
- [SCH,87] Schulmer, Gordon. Handbook of Software Quality Assurance.
U.S.A., Van Nostrand Reinhold Company, 1987.
- "Doce factores de evaluación de herramientas CASE".
En:Computerworld. España, Febrero(1993), pp.23-27.

BIBLIOGRAFIA GENERAL

Acle, Tomasini Alfredo. "Premios de Calidad".
En: Expansión, México, Junio(1993), pp.96.

Adam, Everett E. y Ronald J. Ebert Administración de la Producción y las Operaciones. 4ta edición.
México. Prentice Hall, 1991.

Alfonso, Vicente y Adolfo Blanco. Dirigir con Calidad Total.
Madrid, ESIC, 1990.

Alvarado, Andrés y Ricardo Hernández. Informática en Administración.
México. Ed. Trillas, 1992.

Arrona, Hernández Felipe de J. Herramientas Básicas para Calidad Total. 2a. edición. México, ICASA, 1990.

Aubrey, Charles A. Teamwork: Involving People in Quality and Productivity Improvement.
E.U.A., Quality Press, 1988.

August, Judy H. Joint Application Design.
E.U.A., Yourdon Press Computing Series, 1991.

Bachman, Charlie, "A CASE for Reverse Engineering".
En: Datamation, E.U.A., Julio (1988), pp.52-59.

Bloor Robin, "How iterative prototyping techniques solve problems before applications are implemented".
En: DBMS. E.U.A., Febrero(1993), pp.14-16.

Boone, Gregory, Merlyn Vaughan, Roger Dobratz. The Second Annual Report on CASE.
CASE Research Coporation, 1990.

Buenrostro, Ernesto. Control Total de Calidad a su Alcance.
México, EDIMSA, 1990.

Bursh, John G y Gary Grudnitski. Diseño de Sistemas de Información.
México, Grupo Noriega Editores, 1992.

Corripio, Fernando. Diccionario de Ideas Afines.
España, Ed. Herder, 1985.

Crosby, Philip B. La Calidad no Cuesta.
México, Cía. Editorial Continental, S.A. de C.V., 1990.

Darst, Donald, " Balancing Productivity and Quality".
En: Datamation, E.U.A., (1990).

Edwards, Perry. Systems Analysis & Design.
E.U.A., McGraw-Hill, 1993.

Elmasri, Ramez y Navathe Shamkant B. Fundamentals of Database Systems.
E.U.A., The Benajmin/Cummings Publishing Company Inc., 1989.

Fairley, Richard E. Ingeniería de Software.
México, McGraw-Hill, 1987.

Fisher, Alan S. CASE Using Software Development Tools.
E.U.A., John Wiley & Sons, Inc., 1988.

Foss, W.Burry, " Fast, Faster, Fastest development".
En: Computerworld. E.U.A, Mayo(1993), pp.81-83.

Freeman, Peter. Software Perspectives. The System is the Message.
E.U.A., Addison-Wesley Publishing Company, 1987.

Galgano, Alberto. Calidad Total.
México, Ediciones Díaz de Santos , S.A., 1993.

Gane, Chris. Computer-Aided Software Engineering. The Methodologies, the products, the future.
New York, Rapid System Development Inc., 1988.

Gitlow, Howard S. y Shelly J. Gitlow. Cómo Mejorar la Calidad y la Productividad con el Método Deming.
Colombia, Ed. Norma, 1989.

Gutiérrez, Mario. Administrar para la Calidad.
México, Ed. Limusa, 1991.

Hopeman, Richard J. Administración de producción y operaciones.
México, Cía. Editorial continental, S.A. de C.V., 1992.

Ince, D.C. Ingeniería de Software.
E.U.A., Addison-wesley Iberoamericana, S.A., 1993.

Ishikawa, Kaoru. ¿Qué es el Control Total de Calidad?
Colombio, Ed. Norma, 1986.

Johnson, James R. The Software Factory. Managing Software Development the Maintenance. 2da. edición.
E.U.A., Computer Weekly Publications, 1991.

Maglitta, Joseph. " Michael Hammer, One on One".
En: Computerworld. E.U.A., Enero(1994), pp.84-86.

Jurán, Joseph M. Jurán y el Liderazgo para la Calidad. Manual para Ejecutivos.
Madrid, Ediciones Diaz de Santos, S.A., 1990.

Kendall, Kenneth E. y Julie E. Kendall. Análisis y diseño de sistemas. México, Prentice Hall, 1991.

Korth, Henry F. y Abraham Silberschatz. Fundamentos de Bases de Datos. México, McGraw-Hill, 1987.

Lamprecht, James L. ISO 9000 Preparing For Registration. E.U.A., Marcel Dekker, Inc., 1992.

Laudon, Kenneth C. y Jane Price Laudon. Management Information Systems. New York, Macmillan Publishing Company, 1991.

Martin, James. Information Engineering. Book I, II, III. E.U.A., Prentice Hall, 1990.

Martin, James. Rapid Application Development. New York, McGraw Hill, 1991.

McCabe, Thomas, "Tips On Reengineering Redundant Software". En: Datamation, E.U.A., Abril(1992), pp.71-75.

Moller, Claus, "Calidad Personal Una Inversión". En: Expansión, México, Diciembre(1993), pp.100-102.

Möller, K. H. y D.J. Paulish. Software Metrics. U.S.A., IEEE Computer Society Press, 1993.

Patrinostro, Frank y Steve Ayer. Documenting the Software Development. E.U.A., McGraw-Hill, 1992.

Pereña, Brand Jaime. Dirección y Gestión de Proyectos. Madrid, Ediciones Díaz de Santos S.A., 1991.

Pfleeger, Shari Lawrence. Software Engineering. The Production of Quality Software. 2a.edición.
Canadá. Macmillan Publishing Co., 1991.

Pozo, Pino Jaime. Control Total de Calidad.
México, Editorial Font, S.A., 1990.

Powers, Michael J, Paul H. Cheney, Galen Crow. Structured System Development.
E.U.A., Boyd & Fraser Publishing Company, 1990.

Pressman, Roger S. Ingeniería del Software. Un Enfoque Práctico. 3a.edición.
México, McGraw-Hill, 1993.

Randolph, Alan y Barry Z. Posner. Las Diez Reglas de Oro para Trabajar en Equipo.
México, Ed.Grijalbo, 1991.

Ruelas Alejandro, "Calidad se Escribe con C de Competitividad".
En: Computerworld. México, Febrero(1994), pp.33-34.

Sánchez, Sánchez Antonio. La inspección y el control de calidad.
México, Ed. Limusa, 1988.

Schulmeyer,G. Gordon. Handbook of Software Quality Assurance.
U.S.A., Van Nostrand Reinhold Company, 1987.

Schulmeyer, G. Gordon. Zero Defect Software.
E.U.A., McGraw Hill, 1990.

Shecter, Edwin S. Managing For World-Class Quality.
E.U.A., Quality Press, 1992.

Sommerville, Ian. Ingeniería de Software.
México, Addison-Wesley Iberoamericana, S.A., 1988.

Tomasini, Alfredo Aclé. Planeación Estratégica y Control Total de Calidad.
3a. Edición.
México, Editorial Grijalbo S.A., 1990.

Udaono, Miguel Durán. Gestión de Calidad.
Madrid, Ed. Díaz de Santos, 1992.

Walton, Mary. Cómo Administrar con el Método Deming.
México, Ed. Norma, 1990.

Yourdon, Edward. Análisis Estructurado Moderno.
México, Prentice Hall, 1993.

ANEXO A

A continuación se presenta una lista de las herramientas CASE más actuales en el mercado que soportan una o más fases del ciclo de vida de desarrollo de sistemas, agrupadas en las tres categorías de herramientas que existen de CASE, haciendo además mención a la plataforma o sistema operativo en el cual pueden trabajar.

UPPER CASE

Compañía	Producto	Plataforma
AD/Consultants Inc.	Ad/Modeller	Windows, DOS
AGS Management Systems Inc.	FirstCASE	Windows, DOS
American Management Systems Inc.	Life-Cycle Productivity System	DOS, OS/2
Andersen Consulting	Plan/1	OS/2, PM
	Plan/1 For Windows	Windows, DOS
Applied Business Technology Corp.	Project Bridge	DOS
Ascent Logic Corp.	RDD-100	Macintosh, UNIX, Wkst
Bachman Information Systems Inc.	Data Analyst	DOS
Cadre Technologies Inc.	Teamwork/IM	OS/2, VAX/VMS, UNIX, Wkst
CGI Systems Inc.	LifeCycle Manager	Windows, DOS, MVS, Unisys, Tandem, UNIX
Chicago Computer Works	Personal CASE	DOS
	Personal CASE For Windows	Windows, DOS
Computer Associates International Inc.	CA-ADVISOR	DOS
	CA-ESTIMACS	DOS
	CA-PLANMACS	DOS
	CA-UNIPACK/PEP	DOS
Denver Metrics Group	Software Metrics Repository	DOS
Evergreen CASE Tools Inc.	EasyCASE Plus	DOS
Holland Systems Corp.	PROdeveloper	DOS, MVS
IBM	DevelopMate	OS/2, MVS
Iconix Software Engineering Inc.	CoCoPro	Macintosh
Intersolv Inc.	PC Prism	DOS
Kartech Data Services Inc.	PC-Proto III	DOS
Manager Software Products	DesignManager	MVS, VM
	ProjectManager	MVS
Meta Software Corp.	MetaDesign 3.0	Macintosh
	MetaDesign For Windows	Windows, DOS
Nichols & Co.	ORO Project Management	DOS, Wkst, AS/400
Popkin Software and Systems Inc.	System Architect	OS/2, PM
	System Architect for Windows	Windows, DOS

Productivity Management Group Inc.	Productivity Manager	DOS
Quantitative Management Inc.	Software PADS	DOS
	SEAS	DOS
	SLIM	DOS
Rational	RationalRose	Wkst
Spectrum International Inc.	Spectrum/Manager	DOS
SunPro	NSE	UNIX, Wkst
Synon Inc.	Synon/ENTRY	AS/400

MIDDLE CASE

<i>Compañía</i>	<i>Producto</i>	<i>Plataforma</i>
AD/ Consultants Inc.	Ad/Modeller	Windows, DOS
Adaparc Corp.	DF/DP	MVS
Advanced Logical Software Inc.	AnaTool 3.1	Macintosh
	Blue/80	Macintosh
Advanced Systems Technology Corp.	Revengg	VAX/VMS, UNIX, Wkst
	SafeSpan	DOS, VAX/VMS, Wkst
Allen Systems Group Inc.	Brief-Case	MVS
American Digital Technologies Inc.	DEST CASE Tool	Macintosh, VAX/VMS
Andersen Consulting	Design/1	DOS, OS/2, PM
	Design/1 For Windows	Windows, DOS
D. Appleton Co. Inc.	IDEF/ Leverage	VAX/VMS, MVS
	ModelPro 3.5	Macintosh
	ModelPro for Windows	Windows, DOS
Ascent Logic Corp.	RDD-100	Macintosh, UNIX, Wkst
Aurastar Information Systems Inc.	Albert SCADA	DOS
Bachman Information systems Inc.	Bachman/Analyst	OS/2, PM
	Bachman/Designer For CSP	DOS
	Data Analyst	DOS
	Database Administrator	DOS, OS/2
	DBA Catalog Extract	DOS
M. Bryce & Associates Inc.	Pride-DBEM	OS/2, VAX/VMS, MVS, AS/400
	Pride-ISEM	OS/2, VAX/VMS, MVS, AS/400
Cadre Technologies Inc.	Teamwork For OS/2	OS/2
	Teamwork/Access	OS/2, VAX/VMS, UNIX, Wkst
	Teamwork/Ada	VAX/VMS, Wkst
	Teamwork/IM	OS/2, VAX/VMS, UNIX, Wkst
	Teamwork/PCSA	DOS
	Teamwork/SA	OS/2, VAX/VMS, UNIX, Wkst
	Teamwork/SIM	Wkst
Cadware Inc.	System Developer I	DOS, OS/2
	System Developer II	DOS, MVS
Caine Farber & Gordon Inc.	PDL/81	DOS, OS/2, VAX/VMS
Cascade Software Systems Inc.	Lyddia	DOS
Caseworks Inc.	CASE:W Corporate Edition	Windows, DOS
CGI Systems Inc.	DesignAid II	DOS

Compañía	Producto	Plataforma
Chen & Associates	ER-Designer	Windows, DOS
	Normalizer	Windows, DOS
	SchemaGen	Windows, DOS
Chicago Computer Works	Personal CASE	DOS
	Personal CASE for Windows	Windows, DOS
Comdisco Systems Inc.	SPW	VAX/VMS, UNIX, Wkst
Computer Associates International Inc.	CA-AD/VANCE DataModeler	MVS
	CA-DATACOM/CONCEPTOR	DOS
	CA-DB/CONCEPTOR	DOS
	CA-DB-ARCHITECT	DOS
	CA-IDMS/CONCEPTOR	DOS
Computer Science Innovations Inc.	SDL	DOS
Computer Systems Advisers Inc.	POSE	DOS, UNIX
	Silverrun	Macintosh, OS/2, PM
	Silverrun For Windows	Windows, DOS
Compuware Corp.	Pathvu	DOS, MVS, Unysis
Digital Equipment Corp.	DECdesign	VAX/VMS
Excel software	MacAnalyst/Expert	DOS
	MacDesigner	Macintosh
Future Tech Systems Inc.	Envision	DOS, OS/2, PM
	Envision For Windows	Windows, DOS
Holland Systems Corp.	PROdeveloper	DOS, MVS
ICAD	ICAD System	Wkst
Iconix Software Engineering	FreeFlow	Macintosh, DOS, UNIX, Wkst
	FreeFlow for Windows	Windows, DOS
	AdaFlow	Macintosh
	DataModeler	Macintosh
	Quickchart	Macintosh
	Smartchart	Macintosh
ICS Software Group Ltd.	System Control	Bull
Impulse Engineering	Documenter/Diagrammer	DOS
Information Engineering Systems Corp.	IE: Expert	Windows, DOS
Informix Software Inc.	OpenCASE	UNIX
Intersolv Inc.	Exceleator II	DOS, OS/2, PM, VAX/VMS, Wkst
	Exceleator/RTS	DOS, VAX/VMS
	VSDesigner	Designtool

Compañía	Producto	Plataforma
Kartech Data Services Inc.	PC-Proto III	DOS
	ADW/MVS	MVS
	ADW/Analysis Workstation	OS/2, PM
	ADW/Design Workstation	OS/2, PM
LBMS Inc.	Systems Engineer	DOS
	Erwin/ERX	DOS
Manager Software Products Inc.	DesignManager	MVS, VM
	ManagerVIEW	Windows, DOS
	SourceManager	MVS, VM
Mark V. Systems Inc.	Adagen	DOS, OS/2, VAX/VMS, UNIX, Wkst, DG
	Adagen For Windows	Windows, DOS
	Object Maker	DOS, OS/2, VAX/VMS, UNIX, Wkst, DG, PM
	Object Maker for Windows	Windows, DOS
McCabe & Associates Inc.	Design Complexity Tool	DOS, VAX/VMS, UNIX, Wkst
Menio Business Systems Inc.	Foundation Vista	Macintosh, Tandem
Mentor Graphics Corp.	Auditor	VAX/VMS
	Case Bench	VAX/VMS, UNIX
	Case Station	VAX/VMS, Wkst
	CodeLink Station	Wkst
	Designer	VAX/VMS, UNIX
Meridian Software Systems Inc.	OpenSelect CASE	DOS
Meta Software Corp.	Design/CPN	Macintosh, Wkst
	Design/IDEF	Macintosh
	Design/IDEF For Windows	Windows, DOS
	Design/OA	Macintosh, Wkst
	Design/OA For Windows	Windows, DOS
	MetaDesign 3.0	Macintosh
	MetaDesign For Windows	Windows, DOS
MT. St. Helens Software	Stage #1	DOS
Multiprocessor Toolsmiths Inc.	CASEworks/RT	UNIX
Netron Inc.	NETRON/CAP	DOS, OS/2, PM, VAX/VMS, MVS, VM, Wang
ObjectCraft Inc.	ObjectCraft	DOS
Paradigm Systems Inc.	Locate	DOS
The Progeni Corp.	MicroCharter	DOS
Protosoft Inc.	Object Plus	Windows, DOS, Wkst
Quantitative Management Inc.	Software Schooner	DOS

<i>Compañía</i>	<i>Producto</i>	<i>Plataforma</i>
Scandura Intelligent Systems	PRODOC re/Nu Sys W-b	DOS, UNIX, Wkst
	re/NuSys Designer	DOS, Wkst
	re/NuSys H-L Design	DOS, Wkst
Science Applications International Corp.	SAIC- SDDL	DOS, VAX/VMS, UNIX, MVS, wKST, uNI SYS, DG
Scientific and Engineering Software Inc.	SES/Workbench	UNIX, Wkst
Silico-Magnetic Intelligence Corp.	Top-Down Designer	DOS
Six Sigma CASE Inc.	Canonizer	DOS, UNIX, Wkst
Software AG of North America	Natural Architect Workbench	Macintosh
Software Originals	MacStile	Macintosh
Software System Design Inc.	AISLE/QualGen	DOS, VAX, VMS, UNIX, Wkst, DG, Concurrent, Pyramid
	CISLE/CRIS	DOS, VAX, VMS, UNIX, Wkst, DG, Concurrent, Pyramid
	CISLE/QualGen/C	DOS, VAX, VMS, UNIX, Wkst, DG, Concurrent, Pyramid
	CISLE/TestGen/C	DOS, VAX, VMS, UNIX, Wkst, DG, Concurrent, Pyramid
South-Western Publishing Co.	BriefCASE	DOS
SPS Software Products & Services Inc.	EPOS	DOS, VAX, VMS, UNIX, Wkst, DG, Concurrent, Pyramid, MVS
SQL Solutions Inc.	Deft	Macintosh, VAX/VMS, UNIX, Tandem
	Personal Deft	Macintosh
StarSys Inc.	MacBubbles	Macintosh
Sterling Castle Software	Logic Gem	DOS
StructSoft Inc.	TurboCASE 4.0	Macintosh
	Synon/2E	AS/400
	Synon/ENTRY	AS/400
Transform Logic Corp.	DesignAid II	DOS
Viewlogic Systems Inc.	Express VHDL	DOS, VAX/VMS, UNIX, Wkst
Visible Systems Inc.	Visible Analyst	DOS
	Visible Analyst Workbench	DOS
Yourdon Inc.	Analyst/Designer Toolkit	DOS

LOWER CASE

<i>Compañía</i>	<i>Producto</i>	<i>Plataforma</i>
Advanced Software Automation Inc.	Hindsight 2.0	DOS,VAX/VMS, UNIX, Wkst
Bachman Information Systems Inc.	Analyst Capture for COBOL	OS/2, PM
	Analyst Capture for IMS	OS/2, PM
	Analyst Capture for PL/1	DOS, OS/2
	Data Analyst Capture	DOS
	Reengineering Product Set	DOS, OS/2, MVS
Busines Computer Design	ProGen Plus	AS/400
Cadre Technologies Inc.	PathMap	DOS
	Teamwork/C Rev	VAX/VMS, Wkst
	Teamwork/FORTRAN Rev	Wkst
Caseworks Inc.	CASE:PM For C	OS/2, PM
	CASE :PM for COBOL	OS/2, PM
CGI Systems Inc.	Source/RE	DOS
	PacReverse	MVS
Comdisco Systems Inc.	SPW	VAX/VMS, UNIX, Wkst
Compuware Corp.	DATATEC	MVS
	Retrofit	DOS, MVS, UNISYS
Cortex Corp.	Cortex Documentor	VAX/VMS
Gemma International Inc.	Lansa and Garnet	AS/400
Hewlett-Packard Co.	VEE-Engine	Wkst
IBM	DevelopMate	OS/2, MVS
Information Builders Inc.	FACT	OS/2, PM
Informix software Inc.	OpenCASE	UNIX
Integrated Systems Inc.	AutoCode	VAX/VMS, Wkst
InterPort Software Corp.	InterCASE Reverse Engineering Workbench	DOS, VAX/VMS,UNIX, MVS
	InterCycle	OS/2, PM
Intersolv Inc.	APS Developer Center	DOS, MVS,VM
	Design Recovery for DOS	DOS
	XL/Recover	DOS
JBA International Inc.	IPG 400	AS/400
KnowledgeWare Inc.	ADW/Construction Workstation	OS/2, PM
	IEW/Construction Workstation For OS/2	OS/2
Lawson Inc.	Lawson Universe/CASE	UNIX, Wkst, Unysis
LBMS Inc.	SE/Construction	MVS
Manager Software Products Inc.	SourceManager	MVS, VM
McCabe & Associates Inc.	Battlemap Analysis Tool	DOS, VAX/VMS, UNIX, Wstk

Compañía	Producto	Plataforma
Multi Soft Inc.	EasySAA	DOS, OS/2
Netron Inc.	NETRON/CAP	DOS, OS/2, PM, VAX/VMS, MVS, VM, Wang
Nu Thena Systems	Foresight	VAX/VMS, UNIX, Wkst
Object Craft Inc.	ObjectCraft	DOS
PC-Systems Inc.	Model-S	DOS
Promod Inc.	Pro/Source	DOS, VAX/VMS
	Promod/MD	DOS, OS/2, VAX/VMS, UNIX, Wkst.
Reasoning Systems	Refine Language tools	Wkst
	Refine/FORTRAN	Wkst
Scandura Intelligent Systems	re/NU Sys Pseudocode	DOS, Wkst
	re/NuSys Reverse Engineer	DOS, Wkst
SEEC Inc.	SEEC/Care COBOL Analyst	Windows, DOS
Silico-Magnetic Intelligence Corp.	Better-C	DOS
Software AG of North America	Natural Construct	VAX/VMS, MVS, VM, Wang
Software Eclectics Inc.	SE/One	DOS
Software Systems Design Inc.	AISLE/ADADL	DOS, VAX/VMS, UNIX, Wkst, DG, Concurrent.
	AISLE/GraftGen	DOS, VAX/VMS, UNIX, Wkst, DG, Concurrent.
	CISLE/GrafGEN/C	DOS, VAX/VMS, UNIX, Wkst, DG, Concurrent.
Sterling Software Inc.	Mark IX	DOS, MVS, VM
Synthesis Computer Technologies Inc.	CASE/AP	VAX/VMS, UNIX, DG
System Software Associates Inc.	AS/NET	AS/400
	AS/SET IWS	OS/2, PM
TLA Systems and Education Ltd.	Brackets	DOS
UES Inc.	Ki Shell	DOS, VAX/VMS, UNIX
Unisys Corp.	Linc II	Unysis
Varnet Corp.	Power Tools	UNIX
Viasoft Inc.	Via/Renaissance	MVS
Visible Systems Corp.	Visible Prototyper	DOS

ANEXO B
CUESTIONARIO 1

HERRAMIENTAS CASE EN LAS EMPRESAS DE MTY.**CUESTIONARIO.**

Este cuestionario tiene como objetivo conocer si el área de desarrollo de sistemas de su empresa está siendo soportada por herramientas CASE, en dado caso que usted no conozca este tipo de herramientas conteste sólo las preguntas 1 y 2.

1. Nombre de la empresa.

2. Nombre de la persona que proporcionó la información, indicando el puesto que desempeña.

3. ¿Qué metodologías, técnicas o herramientas CASE utilizan en su empresa?

4. Nombre de la herramienta.

5. Compañía que la distribuye.

6. En qué tipo de computadora o plataforma corre su herramienta.

7. Configuración necesaria.

8. Costo.

9. Última versión

10. ¿Qué clase de metodología y técnica soporta?

11. ¿Puede la herramienta ser modificada por el usuario?

12. Limitaciones gráficas de la herramienta.

13. ¿Qué tipo de documentación genera?

14. ¿La herramienta soporta la administración de proyectos?

15. El proveedor de la herramienta le proporciona el soporte técnico y la capacitación.

16. Les han informado si habrá nuevas versiones de la herramienta.

17. Funciones o características adicionales de la herramienta.

CUESTIONARIO 2

Encuesta sobre el uso de metodologías, técnicas y herramientas en el área de desarrollo de sistemas.

Objetivo de la encuesta.

- Conocer el uso que le dan a las herramientas CASE.
- Conocer los beneficios y fracasos que representa utilizar este tipo de herramientas.
- Tener más fundamentos para asegurar el cómo lograr una calidad en el desarrollo de sistemas con CASE.
- Dar a conocer el tipo de cambio cultural en las personas al introducir CASE.
- Dar a conocer también el grado de preparación que se requiere para el uso de esta tecnología que apoya al desarrollo de sistemas.

Una vez que se obtenga la información de diferentes empresas, ésta será confidencial reportando a cada empresa sólo resultados en forma global. Esta información será de gran utilidad a las empresas porque conocerán:

- Las metodologías, técnicas y herramientas más actualizadas.
- El cómo mejorar la calidad en el desarrollo de sistemas.
- Beneficios y fracasos en el uso de CASE.
- Aquellas metodologías que benefician el proceso de desarrollo de sistemas.
- El perfil del departamento de informática para utilizar CASE.

Especificaciones del llenado de la encuesta

Las encuestas serán entregadas en varias empresas al responsable de desarrollo, líder del proyecto o responsable del área de sistemas, quien será el indicado para decir quiénes pueden ser las personas más idóneas para contestar este cuestionario.

Pueden sacarse varias copias al cuestionario de tal manera que éstos se entreguen a diferentes personas del departamento para que contesten aquellas secciones o preguntas con las que más estén familiarizados o en las cuales sean expertos.

Es importante que se conteste este cuestionario en su totalidad para llevar a cabo una mejor evaluación y retroalimentación.

Para contestarse tomar en cuenta lo siguiente:

- Donde existan líneas " _____ " favor de llenarse.
- Si la pregunta tiene varias respuestas seleccionar sólo una(la que más se adecue).
- Si contesta otros especificar cuales.
- En las tablas deberá realizarse el llenado considerando la pregunta por renglón y respondiendo para cada columna.
- Donde existen corchetes [] marque con una cruz la respuesta más apropiada.

Si considera que es poco el espacio para responder a una pregunta determinada anexas la respuesta en las hojas finales de esta encuesta.

Les agradeceríamos mucho su colaboración y además que el cuestionario estuviera contestado en 12 días aproximadamente a partir de la fecha de la entrega.

Si usted tiene alguna duda de este cuestionario no dude en comunicarse al 3-38-50-50 ext 215 0 201 con la Ing. Ma. del Consuelo Jiménez Fdz. del departamento de Ciencias Computacionales de la Universidad de Monterrey.

La información que se obtenga de este cuestionario será de un gran valor como apoyo a nuestros 4 cursos de desarrollo de sistemas en la carrera de Ingeniero en Sistemas Computacionales , y también para el proyecto "Desarrollo e implantación de un sistema con calidad " para obtener el grado de Maestría.

Agradezco de antemano la atención prestada y quedo de usted:
Atentamente.

Ing. Ma. del Consuelo Jiménez.
Depto. Ciencias. Computacionales
Universidad de Monterrey

I. Perfil de la Compañía/Organización.

1. Responsable de llenado de cuestionario.

Nombre: _____

Puesto: _____

Teléfono: _____

2. Nombre de la empresa: _____

3. Tipo de industria:

_____ Banquera, Financiera, Aseguradora, Crédito.

_____ *Manufactura.*

_____ Gobierno.

_____ Educación.

_____ Comerciante: Mayoreo y Menudeo.

_____ Servicios: Médicos, Salud, Legales, Profesional.

_____ Servicio Público.

_____ Servicios de Computación.

_____ Agricultura, Minería, Construcción.

_____ Otros. _____
Especifique

4. Ingresos brutos por año. (En nuevos pesos).

30-100 100-500 500-1000 1000-1500 Más de 1500

5. Número de empleados.

Menos de 50 50-100 100-500 500-2500 2500-7500 7500-50000
Más de 50000

6. Número de empleados en el departamento de informática.

1-5 5-10 10-15 15-20 20-50 50-100
Más de 100.

7. Número de empleados destinados al desarrollo de sistemas.

Planeación _____ Diseño _____ Pruebas _____
Análisis _____ Construcción _____ Mantenimiento _____

8. ¿Cuenta usted con personal dedicado a investigación tecnológica sobre nuevas, metodologías, técnicas y herramientas en el desarrollo de sistemas? ¿ Cuantos son? _____

II. Antecedentes del departamento de sistemas.

9. El sistema operativo para el medio ambiente de producción.
(1-Primario, 2- Secundario, etc)

_____ DOS/VSE	_____ MVS o MVS/XA
_____ AS/400	_____ VM
_____ UNIX	_____ DEC/VMS
_____ MS-DOS	_____ OS/2
_____ Otros _____	

Especifique

10. El sistema operativo del medio ambiente de desarrollo.
(1-Primario, 2-Secundario, etc)

_____ DOS/VSE	_____ MVS o MVS/XA
_____ AS/400	_____ VM
_____ UNIX	_____ DEC/VMS
_____ MS-DOS	_____ OS/2
_____ Otros _____	

Especifique

11. Principales sistemas administradores de bases de datos(DBMS).
(Indicar el lugar que le corresponde 1ero. 2do. 3ero).

_____ IMS	_____ DB2
_____ SUPRA	_____ ULTRA
_____ ADABAS	_____ INGRES
_____ IDMS	_____ Datacom DB
_____ ORACLE	_____ Model 204
_____ Informix	
_____ SQL/DS	
_____ Otros _____	

12. Principales lenguajes de desarrollo .
(Indica el lugar que les corresponde 1ero. 2do. 3ero).

_____ COBOL	_____ PL/1
_____ FORTRAN	_____ C
_____ RPG	_____ ADA
_____ 4GL _____	

Especifique

_____ OTROS _____

13. ¿Cuál es la proporción de PC's, estaciones de trabajo o terminales para los desarrolladores?
(Ejemplo 1 PC: 2 Desarrolladores)

_____ 1:1 a 1:3	_____ 1:7 a 1:9
_____ 1:4 a 1:6	_____ 1:10 o más.

14. En la práctica que es lo más importante su tu organización:

- Mejorar la productividad.
- Mejorar la calidad.
- No se había pensado en calidad y productividad sólo desarrolla para satisfacer al usuario en el momento que lo requiera.

15. Si su organización tiene un programa de mejora de calidad, por favor marque aquél que mejor caracterice ese programa.

- No existe un programa de mejora de calidad que se este utilizando en este momento.
- Uso de slogans.
- Enfoque sobre pruebas e inspecciones.
- Enfoque sobre prevención de defectos.
- Enfoque sobre mejorar el proceso y utilizar técnicas de control de proceso estadísticas.

16. ¿Sigue alguna filosofía de calidad de autores como Deming, Crosby, etc. enfocada al desarrollo de sistemas?

Si No
 _____ De quién.

III. El uso de las metodologías.

<i>Métodos estructurados</i>			<i>Metodologías del ciclo de vida</i>	
BAC - Bachman	HOL - Holland	WML - Ward-Mellor	DCE-DCE	SDM-SDM/70
BHY - Bacing-Hatley	JSN- Jackson	WOR-Warnier/Orr	DSD-DSSD	SDT-SADT
CHN - Chen	MAR-Martin	YDM-Yourdon/ DeMarco	IFE-Ing.de Inf.	SPM-SPECTRUM
CDT- Codd/Date	MER-Merise	INH- In-house	LSM-LSDM	STR-STRADIS
CON-Constantine	FIN-Finkelstein	GSN-Gane&Sarson	MET-Method/1	TIP- TIP
SHN-Shir Nijssen Manager.	OTH-Otros _____		MMR-METHOD	
			MIN-Mini-Asyst	
			PRD-PRIDE	
			INH- In-house	
			OTH-Other _____	

Utilice los códigos anteriores para llenar al tabla.

Métodos Estructurados

**Metodologías del
Ciclo de vida**

<p>17. ¿Cuáles métodos son regularmente utilizados? Indique el orden en cuanto a dominio o uso.</p>	<p>Primero</p>	<p>Segundo</p>	<p>Tercera</p>		
<p>18. ¿Qué tanto tiempo tiene la organización utilizando estos métodos? (meses)</p>					
<p>19. ¿En cuánto tiempo los analistas/ desarrolladores y programadores/ se entrenan en el uso de estos métodos.</p>					
<p>20. ¿Cuánto tiempo los analistas/desarrolladores y programadores/ están regularmente utilizando estos métodos?</p>					
<p>21. ¿Cómo ud. espera que el uso de estos métodos cambie en el próximo año? Responda por cada método "+": Incrementa, "0" permanece igual, "-": decrementa</p>					

Métodos Estructurados**Metodologías del
Ciclo de vida**

22 ¿Estos métodos se mejoran o se actualizan? (Escala 1-nunca, 5-frecuentemente)	Primero	Segundo	Tercero		
23. En una escala de 1- 5 (1-bajo, 5-alto) ¿Cuál sería la visión de la organización de el éxito en el uso de éstos métodos?					
24. Aproximadamente que porcentaje de nuevos trabajos de desarrollo están empezando a soportarse con estos métodos					
25. Aproximadamente que porcentaje de trabajo de mantenimiento y mejoras está siendo soportado con esos métodos.					
26. Los métodos que utilizas ¿que fases de ciclo de desarrollo de sistemas soportan?					

27. En la escala de 1 a 5 (1-no lo soporta, 5-alto soporte) que tan bien los métodos son soportados por herramientas CASE si su organización las está utilizando actualmente.					
---	--	--	--	--	--

28. Las siguientes preguntas están relacionadas con la experiencia de tu organización en relación a las herramientas CASE. Si no tiene herramientas CASE después de contestar las siguientes preguntas y la sección IV, salte a la sección VI.

- Se está recopilando información sobre algunas herramientas.
- Se está en proceso de evaluación de alguna herramienta.
- No se conoce nada acerca de ellas no se piensa utilizarlas.

IV.La problemática en el desarrollo de sistemas.

29. Indica en cada etapa del desarrollo de sistemas cuáles han sido los diferentes problemas que se han presentado, sus causas, la solución que le han dado y el impacto que ha tenido esa solución. Asimismo indique según su experiencia quien es el personal más indicado para ser involucrados, responsables y para estar en las revisiones de cada etapa.

Fase		Problemas	Causas	Solución	Impacto
Planeación de sistemas	Involucrados: Responsables: Revisiones:				
Análisis	Involucrados: Responsables: Revisiones:				
Diseño	Involucrados: Responsables: Revisiones:				
Construcción	Involucrados: Responsables: Revisiones:				

Pruebas	Involucrados: Responsables: Revisiones:				
Implementación	Involucrados: Responsables: Revisiones:				
Mantenimiento	Involucrados: Responsables: Revisiones:				

V. Experiencia en CASE.

	Herramientas Front-end CASE Para planeación estratégica, Análisis, Diseño lógico.	Herramientas Back-end CASE Herramientas para diseño físico, construcción y re-Ingeniería.
30. Aproximadamente qué porcentaje de nuevos trabajos de desarrollo están empezando a soportarse con estos métodos.		
31. Aproximadamente qué porcentaje de trabajo de mantenimiento y mejoras está siendo soportado con esos métodos.		
32. ¿Cuánto tiempo de uso? (meses)		
33. Cantidad de desarrolladores entrenados para el uso de la herramienta.		

34. Cantidad de desarrolladores que regularmente utilizan la herramienta.		
35. Indique el éxito con la herramienta en una escala de 1 a 5 (1-bajo éxito-5-alto éxito)		
36. Indique el nombre de cualquier herramienta secundaria que utiliza, si no existe responda "cero".		
37. Indique el uso previo de los métodos estructurados para la introducción de las herramientas CASE en la escala de la 1 a 5 (1-bajo uso, 5 -alto uso)		
<p>38. Que nivel de la organización fue el que impulsó para introducir la herramienta CASE (selecciona sólo una).</p> <p>a) Gerencia. b) Departamento de desarrollo. c) Insistencia por parte del usuario. d) Gerencia y depto de desarrollo. e) Ninguno.</p>		
39. Indica el nombre de la herramienta para el repositorio .	<p>___ Mejora la productividad. ___ Mejora la calidad de sistemas. ___ Se logra cumplir con lo planeado. ___ Vehículo para infundir nuevas metodologías. ___ Mejora la comunicación con los usuarios. ___ Mejora el mantenimiento de aplicaciones. ___ Otros.</p>	<p>___ Mejora la productividad. ___ Mejora la calidad de sistemas. ___ Se logra cumplir con lo planeado. ___ Vehículo para infundir nuevas metodologías. ___ Mejora la comunicación con los usuarios ___ Mejora el mantenimiento de aplicaciones. ___ Otros.</p>
40. ¿Cuáles fueron los beneficios que se lograron al utilizar herramientas CASE en la organización? Indicar 1ero, 2do, 3ero.		
41. Indica que fracasos o problemas hubo con el uso de CASE.		

VI. Su propio punto de vista.

42. Por favor indique abajo aquél que represente el mayor enfoque para mejorar el proceso de desarrollo en tu organización.

(Indique cual 1ero., 2do. y 3ero si es aplicable).

- Prototipo rápido.
- Sesiones utilizando JAD (Joint Application Design)
- Herramientas CASE Front-end.
- Herramientas CASE Back-end.
- Sistemas Expertos.
- Re-ingeniería.
- 4GL's
- Métodos orientados a objetos.
- Métodos estructurados.
- Metodologías del ciclo de vida del proyecto.
- Otros _____

Específica por favor

43. Indique los principales obstáculos para adoptar CASE en tu organización (Indique cual 1ero, 2do, 3ero.)

- Falta de una metodología.
- Resistencia al cambio.
- Falta de integración con las herramientas existentes.
- Baja compatibilidad.
- Baja compatibilidad de CASE con las prácticas de desarrollo existentes.
- Inadecuada disponibilidad de recursos organizacionales.
- Falta de experiencia de los desarrolladores en relación a los métodos estructurados.
- Carencia de un comité administrativo.
- Prueba de valor de que el CASE es inadecuado.
- Falta de un repositorio de IBM estándar.
- Temor del cliente de CASE como una nueva tecnología.

44. Es la filosofía CASE obligatoria a seguir en :

- Toda la organización.
- Departamento.
- Proyecto piloto.
- Es exclusiva de un sólo usuario.
- Otros _____

Especifique

45. _____ Del total de aplicaciones que se han desarrollado cuántas se han producido con herramientas CASE.

46. La capacitación que se les dio fue:

- Por proveedor
- Externa
- No hubo capacitación el desarrollador aprendió conforme se dio el avance del proyecto.

47. ¿Qué cambios en la estructura organizacional han surgido con la introducción y uso de CASE?

48. ¿Cuales han sido? y ¿Qué nuevas funciones se han creado? (Explique).

49. ¿Como se logró el cambio cultural en relación a el uso de herramientas CASE? Se siguió alguna metodología o técnica. (Explica)

50. ¿Qué perfil consideran que debe de tener un analista, el departamento de informática y la empresa para utilizar herramientas CASE?

51. ¿Cree que la herramienta CASE ayuda a mejorar la calidad de los sistemas? ¿ En que porcentaje?

ANEXO C

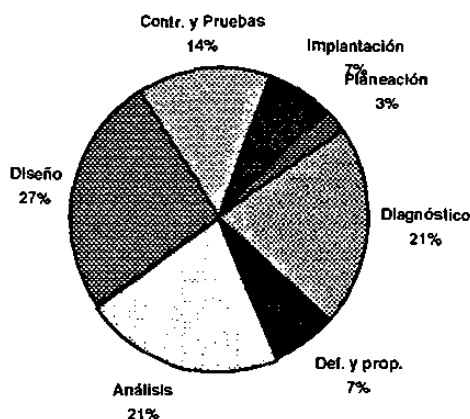
Solicitud de servicio a sistemas	
Dirección o Gerencia: _____ Fecha: _____ Solicitante: _____ Puesto: _____ Teléfono: _____ Ext: _____	
Descripción del problema u oportunidad: _____ _____ _____	
Objetivos: _____ _____	
Beneficios: _____ _____	
Riesgos: _____ _____	
<i>Para uso exclusivo de sistemas</i>	
Solicitud: Aceptada _____ Rechazada _____	
<i>Aceptada</i>	<i>Rechazada</i>
Fecha: _____ Asignado a: _____ Fecha de inicio estimada: _____ Autorizado: _____ <div style="text-align: right;">Firma y nombre</div>	Fecha de aviso al usuario: _____ Rechaza: _____ <div style="text-align: right;">Firma y nombre</div> Razones: _____

Solicitud de mantenimiento a sistemas	
Dirección o Gerencia: _____	Fecha: _____
Solicitante: _____	Puesto: _____
Teléfono: _____	Ext: _____
Tipo de solicitud: Error <input type="checkbox"/> Cambio <input type="checkbox"/> Mejoras <input type="checkbox"/>	
Descripción de la solicitud: _____ _____ _____	
Beneficios o justificación: _____ _____	
Especifique la urgencia: _____ _____	
<i>Para uso exclusivo de sistemas</i>	
Fecha en que se recibió la solicitud: _____	
<p style="text-align: center;"><i>Aceptada</i> <input type="checkbox"/></p> Fecha: _____ Asignado a: _____ Fecha de inicio estimada: _____ Autorizado: _____ <p style="text-align: center;">Firma y nombre</p>	<p style="text-align: center;"><i>Rechazada</i> <input type="checkbox"/></p> Fecha de aviso al usuario: _____ Rechaza: _____ <p style="text-align: center;">Firma y nombre</p> Razones: _____ _____

ANEXO D

Al utilizar herramientas CASE usted obtendrá grandes beneficios, no solo en calidad por permitir una mejor comunicación con el usuario, por generar y mantener la documentación actualizada, obligando además al cumplimiento de estándares; sino por permitir incrementar la productividad del equipo de desarrollo por la facilidad de aprenderlas, de utilizarlas y porque permite al administrador del proyecto o líder tener un control sobre todos los sistemas en desarrollo. Para que todo esto se cumpla es necesario que CASE trabaje en red, es importante además recalcar que CASE reducirá el tiempo que se necesita para desarrollar el sistema más en la etapa de construcción, ayudado a realizar un mantenimiento más rápido por las características que esta herramienta posee.

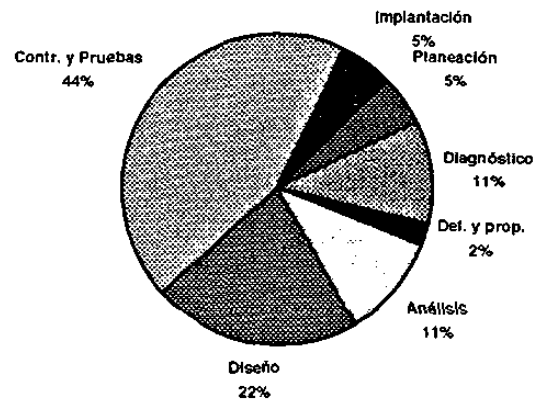
Tomando como referencia información proporcionada por experiencias en el uso de CASE en el desarrollo de sistemas en lo que respecta a tiempos y adaptándolo a la guía propuesta tendríamos que el porcentaje de tiempo aproximado para cada fase si utilizamos CASE sería:



Hay que considerar que estos porcentajes se pueden reducir cuando la herramienta este completamente dominada, ya que resulta ser un poco mayor el tiempo dedicado a las primeras etapas, cuando todavía no se tiene experiencia en este tipo de herramientas.

También si la herramienta puede generar un 100% de código y ser muy poderosa en las pruebas aún más se reducirá el tiempo.

La contraparte sería no utilizar CASE siguiendo la guía teniendo entonces la siguiente distribución:



Con esto vemos que al utilizar herramientas CASE debemos de dedicarle más tiempo a diagnosticar el problema, a definir bien los requerimientos con el usuario y construir la solución a través del un análisis y diseño y no dedicarle más tiempo a la programación o por muchos llamada construcción.

ANEXO E

Compañía	Herramientas que ofrece
Visible Systems Corporation The Bay Colony Corporate Center 950 winter Street Waltham, MA 02514 Tel.(617) 890-2273. Fax.(617) 890-8903 -George G. Cagliusa-	Visible Analyst Workbench 5.0 for Windows.
CGI Systems, Inc. 1180 West Swedesford Road Suite 350, Berwyn PA 19312 Tel.(215) 993-8082 Fax.(215) 993-8125 -Lauren Williams-	PACBASE. PACLAN. Designaid II.
Syscorp International 9430 Research Blvd. Echelon IV Suite 300 Austin TX. 78759 Tel.(512) 338-5800 Fax.(512) 338-5810 -Donald E. Smith-	MicroSTEP
Chen & Associates, Inc. 4884 Consitution Ave. Suite 1-E. Baton Rouge, Louisiana 70808 Tel.(504) 928-5765 Fax.(504) 928-9371 -Jessica T. Eubanks-	Chen-ER-Modeler Workbench

<p>CDSI Compute Data systems, Inc. One Curie Court Rockville, Maryland 20850-4389 Tel.(301) 921-72380 Fax.(301) 948-93280</p> <p>-Heather H. Warren-</p>	<p>Cobol/Metrics. Superstructure/retool. Scan/Cobol. Configure. Sleuth.</p>
<p>Evergreen CASE Tools, Inc. 8522 154 th. Redmond, WA 98052 Tel.(206) 881-5149 Fax.(206) 883-7676</p>	<p>EasyCASE 4.1 for Windows 4.0 for DOS</p>
<p>CADRE technologies Inc. 222 Richmond Street Providence, RI02903 Tel.(401) 351-5950 Ext. 224 Fax.(401) 455-6804</p> <p>-Kelly Bailey-</p>	<p>ObjectTeam. Teamwork.</p>
<p>Viasoft 3033 North 44th Street Phoenix, Arizona 85018 Tel.(602) 952-0050 Fax.(602) 840-4068</p>	<p>ESW</p>
<p>i-Logix Inc. Three Riverside Drive. Andover, MA 01810. Tel.(508) 682-2100 Ext. 3016 Fax.(508) 682-5995 Email. amc@logix.com</p> <p>-Ann-Marie Carroll-</p>	<p>Statemate. ExpressV-HDL.</p>

ANEXO F

HERRAMIENTA ADMINISTRATIVA

Diagrama matricial

Descripción:

Organiza un gran grupo de características, funciones y tareas de forma que los puntos que están lógicamente relacionados entre sí se representan gráficamente.

Usos:

- Relacionar las necesidades de los clientes con las características de los productos y los servicios ofrecidos.

A y B representan los eventos relacionados con el problema

B	b1	b2	bn
A	a1			
	a2			
	.			
	.			
	.			
	an			

En la intersección se analiza la relación.

Fase del desarrollo de sistemas que soporta:

- Planeación.
- Diagnóstico.
- Análisis.

HERRAMIENTA ADMINISTRATIVA

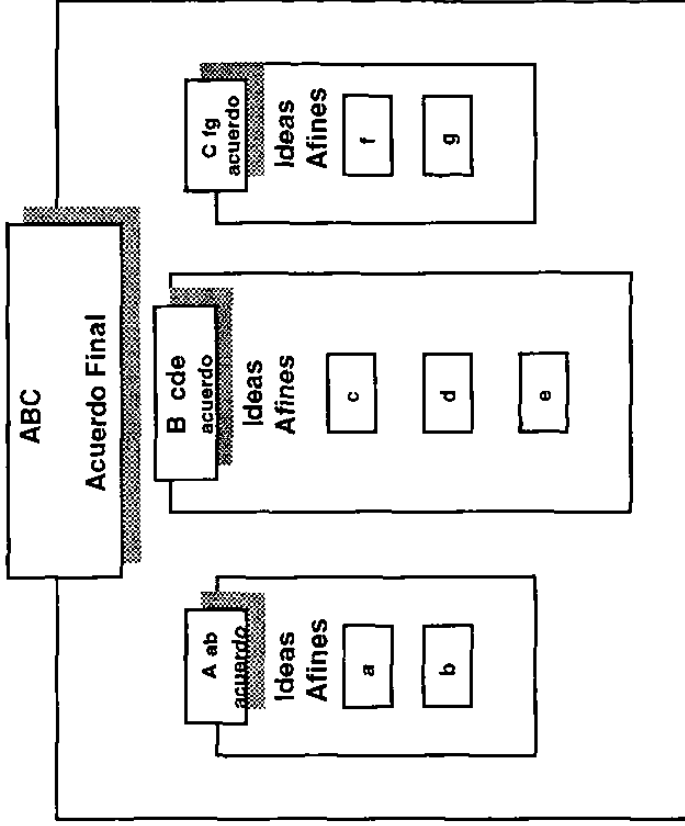
Diagrama de afinidad

Descripción:

Permite estructurar y clasificar ideas poco claras o poco definidas en relación a una situación problemática, integrándolas por afinidad, encontrando así el problema.

Usos:

- Extraer la mayor cantidad de información útil a partir de pocos datos o datos dispersos.
- Comprender y estructurar los problemas poco claros.
- Organizar el pensamiento de un grupo y aclarar su tendencia.
- Aclarar el futuro.
- Organizar y dirigir la experiencia de un grupo hacia la solución de un problema específico.



Fase del desarrollo de sistemas que soporta:

- Planeación.
- Diagnóstico.

HERRAMIENTA ADMINISTRATIVA

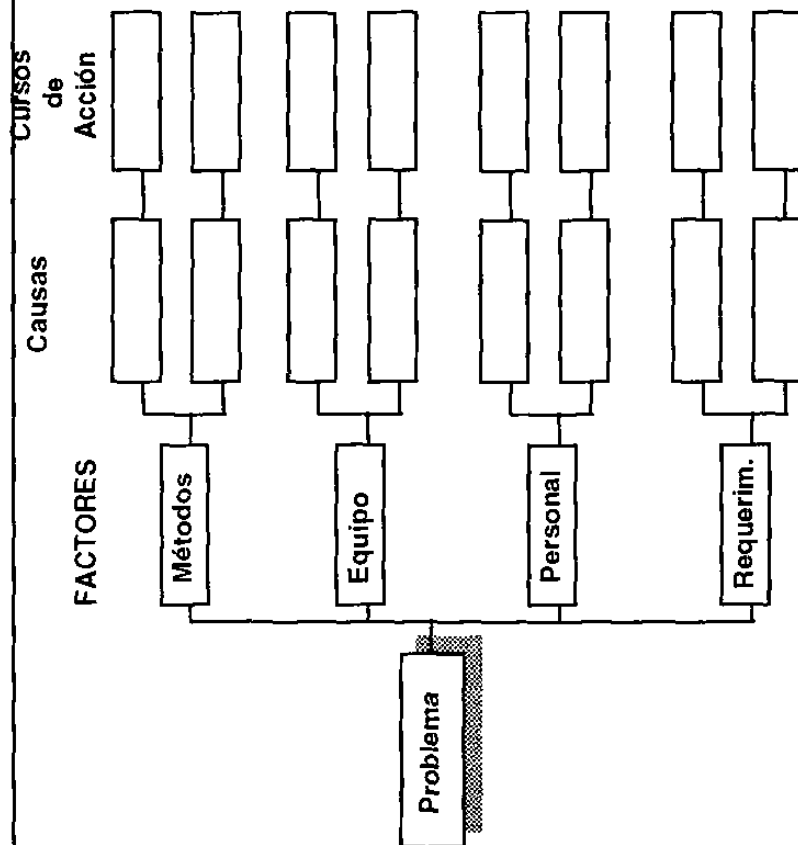
Diagrama de árbol

Descripción:

Define los medios primarios secundarios, etc, y acciones para lograr una meta u objetivo final.

Usos:

- Representación y estructuración de políticas de la empresa.
- El detalle de las actividades de cualquier función empresarial.
- La definición de acciones para la garantía y mejora de calidad.
- Análisis de los deseos de los clientes.



Fase del desarrollo de sistemas que soporta:

- Planeación.
- Diagnóstico.
- Análisis.

HERRAMIENTA ADMINISTRATIVA

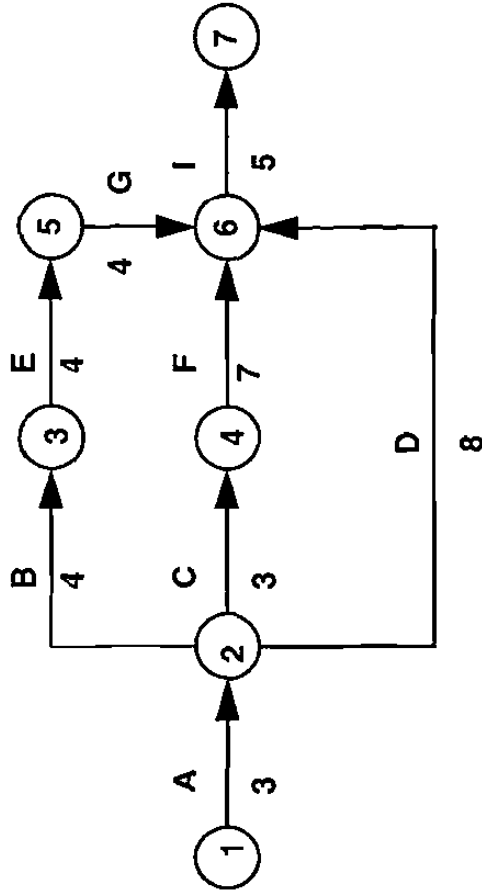
Diagrama de flechas

Descripción:

Se utiliza para representar un programa de trabajo en el cual se requieren varios participantes haciendo uso de una red de actividades y acontecimientos que permite visualizar de forma completa la interdependencia entre las actividades elementales que lo constituyen.

Usos:

- Controlar eficazmente la evolución de los trabajos.
- Optimizar tiempos y procedimientos.
- Indispensable para proyectos a largo plazo.



Fase del desarrollo de sistemas que soporta:

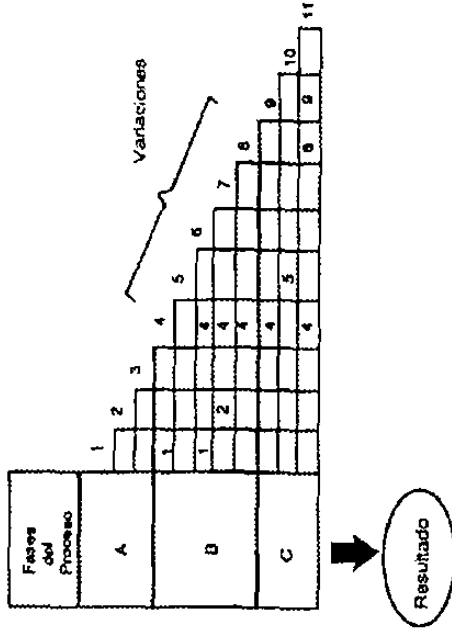
- Planeación.
- Diagnóstico.
- Definición de requerimientos y desarrollo de la propuesta.
- Análisis.
- Diseño.
- Contrucción y pruebas por unidad.
- Pruebas del sistema.
- Implantación.

HERRAMIENTA ADMINISTRATIVA

Análisis matricial de variaciones

Descripción:

Ayuda a identificar variaciones clave en las diferentes fases del proceso, para controlar, asegurando la calidad. Además, previene los errores y las desviaciones en los resultados.



Usos:

- Separar hechos de opiniones.
- Ponerse de acuerdo sobre cómo es que debe de trabajar un proceso y con quién debe interrelacionarse.
- Prevenir las variaciones.

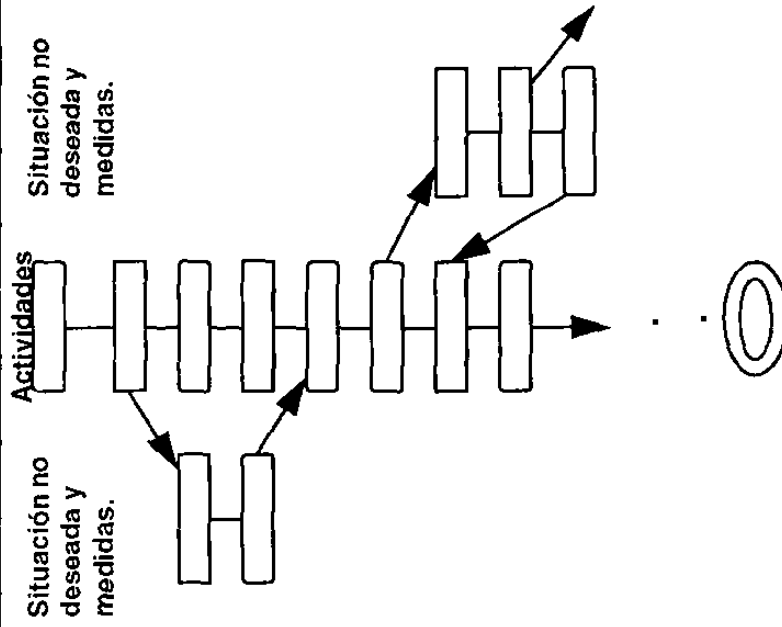
En la intersección se establece la dependencia entre las variaciones.

Fases del desarrollo de sistemas que soporta:

- Puede realizarse antes de empezar el desarrollo de un proyecto.

HERRAMIENTA ADMINISTRATIVA

Diagrama de proceso de decisiones programadas



Descripción:

Programa que muestra el trabajo a realizar con situaciones inciertas y las acciones específicas para contrarrestar lo inesperado.

Usos:

- Se utiliza para elaborar un plan que permita lograr un objetivo final, definiendo los problemas que se presentarían y las acciones apropiadas para así dirigir el evento hacia un resultado deseable.
- Tomar decisiones cuando no se tiene información suficiente.

Fase del desarrollo de sistemas que soporta:

- Puede definirse antes de empezar el desarrollo de un proyecto.
- Planeación.

HERRAMIENTA ADMINISTRATIVA

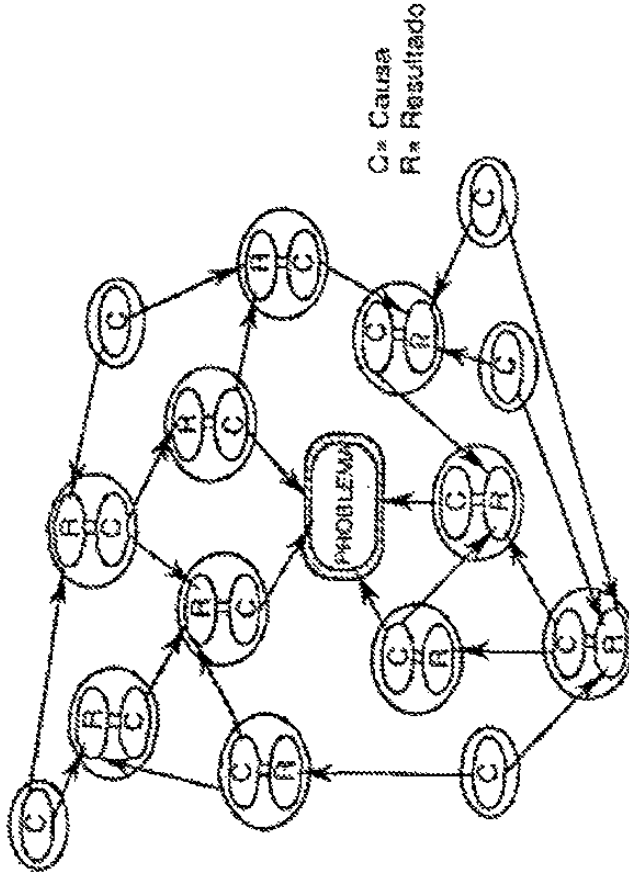
Diagrama de relación

Descripción:

Permite aclarar las causas y sus relaciones para identificar, confirmar y seleccionar las causas originales más importantes que afectan a un problema.

Usos:

- Se puede utilizar cuando las metas de un departamento no se cumplen.
- Cuando no se cumplen tiempos de entrega.
- Siempre que hay errores en el proceso.



Fase del desarrollo de sistemas que soporta:

- Diagnóstico.
- Análisis.

HERRAMIENTA ESTADISTICA

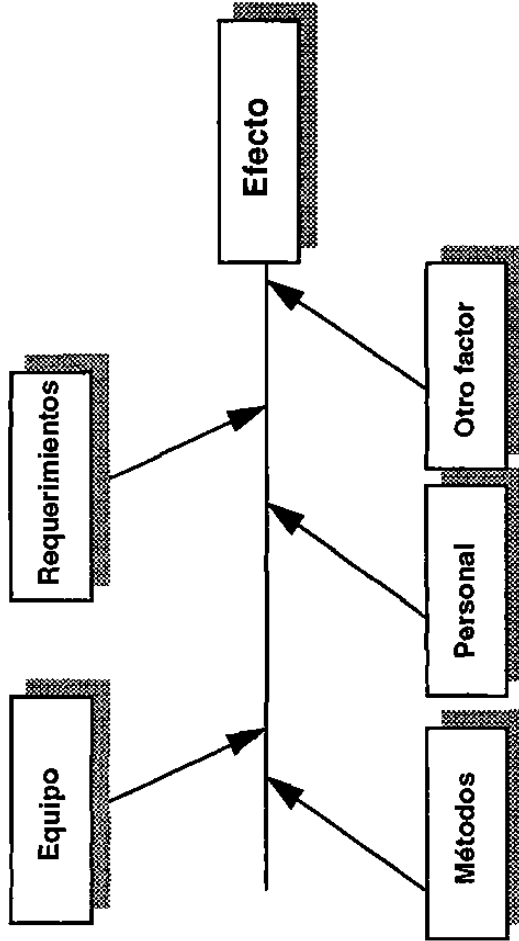
Causa-Efecto

Descripción:

Divide las causas que originan o influyen en cierto problema. El uso de este diagrama facilita el entendimiento y comprensión del proceso y elimina la dificultad del control de calidad, promueve el trabajo en grupo, ya que es necesaria la participación de la gente involucrada en el proceso para su elaboración y uso.

Usos:

- Mejorar la calidad.
- Para el control del proceso.
(Se elabora un diagrama considerando las fases del proceso), especificando los factores vitales a controlar en el proceso, asegurando así las características de calidad del producto.
- Para capacitación de personal para producir calidad.



Fase del desarrollo de sistemas que soporta:

- Diagnóstico.
- Análisis.

HERRAMIENTA ESTADISTICA

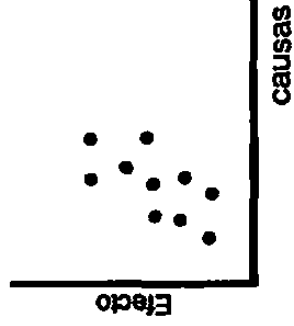
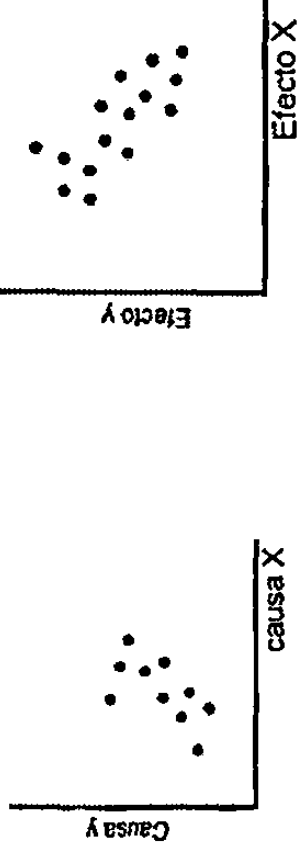
Diagrama de dispersión

Descripción:

Gráfica de puntos que muestra la relación entre un par de puntos dibujados en un par de ejes. Los datos que se grafican son continuos y permiten analizar la relación entre una causa y un efecto, la relación entre una causa y otra causa y la relación entre un efecto y otro efecto.

Usos:

- Confirmar las causas empleando datos que provienen de mediciones.
- Estandarizar factores vitales a controlar en un proceso.
- Determinar la correlación entre dos problemas y poder así seleccionar el más factible de resolver.



Fase del desarrollo de sistemas que soporta:

- Diagnóstico.

HERRAMIENTA ESTADISTICA

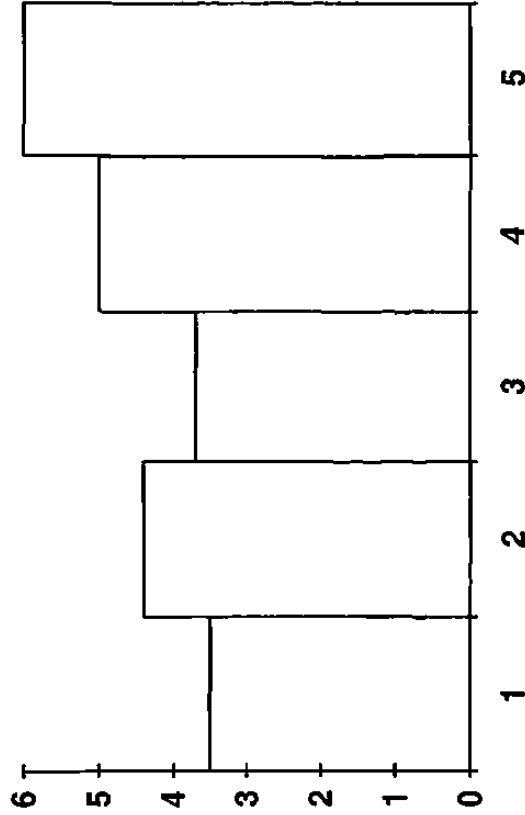
Estratificación

Descripción:

Consiste en clasificar los factores en una serie de grupos con características similares con el fin de encontrar la causa de los problemas. Clasifica datos discretos con el objeto de analizar la causa elegida dada por un diagrama causa-efecto y confirma su efecto sobre la característica de calidad a mejorar o problema a resolver.

Usos:

- Análisis más profundo.
- Confirmar causas de problemas cuando se tenga como base datos discretos.



Fase del desarrollo de sistemas que soporta:

- Diagnóstico.

HERRAMIENTA ESTADISTICA

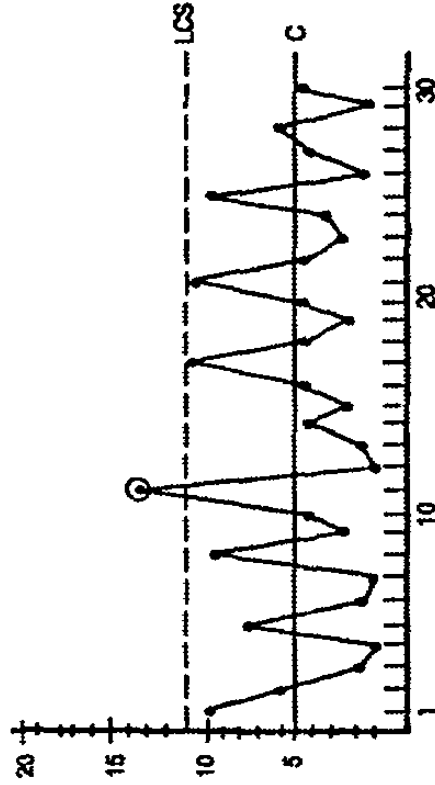
Gráficas de control

Descripción:

Muestra en forma continua la variabilidad de un proceso.
Formada por límites de control (superior e inferior) y haciendo uso de ellos es posible distinguir desviaciones, tanto por causas debidas al proceso o al azar.

Usos:

- Analizar un proceso determinando no sólo su estado sino además si éste está controlado o no.
- Controlar el proceso asegurando su calidad durante la producción.



Fase del desarrollo de sistemas que soporta:

- Diagnóstico.

HERRAMIENTA ESTADISTICA

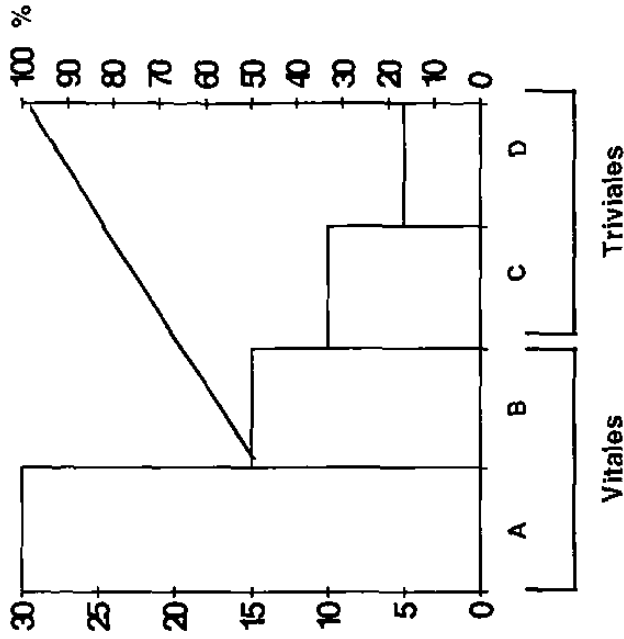
Diagrama de Pareto

Descripción:

Gráfica de barras que clasifica, en forma descendente el tipo de fallas o factores que se analizan en función de su frecuencia o de su importancia absoluta y relativa. Este diagrama clasifica los problemas en orden de importancia, separando aquellos que son vitales de los triviales.

Usos:

- Es el primer paso para efectuar mejoras.
- Ayuda a mejorar la calidad en relación a productos, procesos, servicios, eficiencia, seguridad, ahorro en costos etc.
- Son útiles para confirmar los efectos de las mejoras realizadas



Fase del desarrollo de sistemas que soporta:

- Planeación.
- Diagnóstico.
- Definición de requerimientos.

HERRAMIENTA ESTADISTICA

Hojas de verificación

Descripción:

Para tener control sobre algún proceso se requiere recolectar datos, que sean correctos y que permitan identificar las relaciones que existen entre las causa y efectos con el fin de tener un control en la calidad y productividad, para lo cual es recomendable hacer uso de este tipo de hojas diseñadas con un formato especial.

Usos:

- Verificar productos defectuosos.
- Analizar la localización de defectos.
- Verificar la causa de los defectos.

DEFECTO TIPO	CHEQUEO	SUB TOTAL
A	II	5
B	III	20
C	IV	18
D	V	26
Otros	VI	3
	* GRANTOTAL	75
RECHAZOS TOTAL	UR UR UR UR UR UR UR	57

Chequeo Fase:		
No. Tarea.	Descripción de la tarea	Chequeo
.....	()
.....	()
.....	()
.....	()

Fase del desarrollo de sistemas que soporta:

- Planeación.
- Diagnóstico.
- Definición de requerimientos y desarrollo de la propuesta.
- Análisis.
- Diseño.
- Construcción y prueba por unidad.
- Prueba del sistema.
- Implantación.
- Operación.

HERRAMIENTA ESTADISTICA

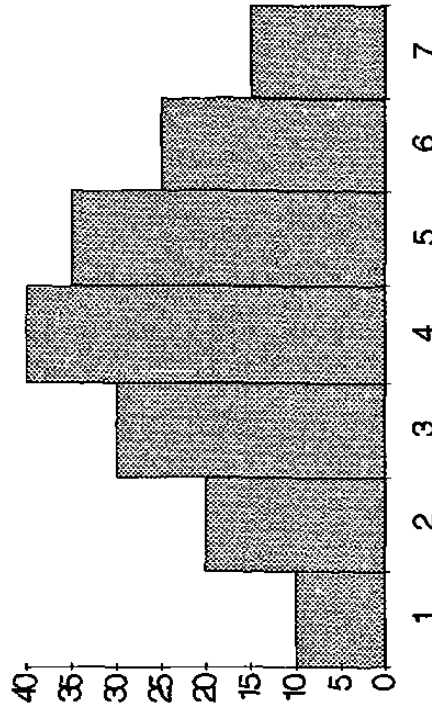
Histograma

Descripción:

Gráfica de barras que presenta los datos en forma agrupada y ordenada con el fin de determinar las veces en que ocurren las variaciones de dichos datos, los datos que representa son continuos, es decir que provienen de mediciones.

Usos:

- Conocer la forma , media y desviación de la distribución del proceso.
- Conocer la relación entre los límites de especificación de distribución del proceso .
- Confirmar los efectos de las mejoras realizadas en el proceso.



Fase del desarrollo de sistemas que soporta:

- Puede ayudar en el diagnóstico.

