

# Universidad Nacional Autónoma de México



**FACULTAD DE ESTUDIOS SUPERIORES**



**“ESTRATEGIA DE CALIDAD PARA EL DESARROLLO  
DE SOFTWARE EN UNA INSTITUCIÓN FINANCIERA”**

**T E S I S  
QUE PARA OBTENER EL TÍTULO DE  
INGENIERO EN COMPUTACIÓN  
PRESENTAN:**

**ALEJANDRO GÓMEZ TORRES  
MARTÍN OROZCO ALEMÁN**

**ASESOR DE TESIS: ING. SILVIA VEGA MUYTOY**

**MÉXICO 2007**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# *Agradecimientos de Alex*

---

*A mi dulce señor:*

*Dios, gracias por la vida tan hermosa que me has dado, por la familia que elegiste para mi, todo aquello que pueda escribir nunca será suficiente porque tu bien conoces mi sentir, a lo largo de los años me enseñaste que la fe, la disciplina y el trabajo constantes son necesarios para alcanzar cualquier meta, he sido afortunado porque siempre has puesto en mi camino todo lo que he necesitado para lograr mis anhelos, significas para mi todo el amor y la razón de mi esencia, de mi existir, gracias por haberme traído hasta aquí, te amo.*

*A mis padres:*

*Con el amor más humilde y sincero les dedico esta obra que sencillamente es el resultado de todos los años de amor, fe y desvelos puestos en su hijo que los ama profundamente, estoy seguro que mi madre Juanita así como mi padre Alex son los seres mas hermosos del universo porque siempre me brindaron todo lo que tuvieron para verme llegar hasta este punto de mi vida, si bien nunca podré regresarles mercedamente todo lo que me han otorgado, al menos déjenme expresarles cuantos los amo y hacerles saber que sin ustedes yo no tendría razón de ser, llegar hasta este punto no fue fácil, pero juntos luchamos contra la adversidad y como pueden ver valió la pena, hoy les obsequio este trabajo solo para darme el gusto de verlos sonreír y recordarles siempre en estas líneas que los llevaré en mi alma eternamente, los amo.*

*A mi hermana:*

*Por ser mi primera amiga y la mejor, por haberme regalado los momentos mas inolvidables de mi infancia, por ser el motivo principal que me ha llevado a ser todo lo que nunca soñé, solo por ti y por el ejemplo que esta obra representa, nunca dejaré de dar todo lo que tengo para mostrarte lo que puedes alcanzar cuando en verdad deseas algo, no olvides que te amo con toda mi alma y que doy gracias a Dios por conocerte y disfrutar de tu compañía, la belleza de la que se estas hecha no tiene comparación, no hay dos como tú, gracias por tus ocurrencias, tus palabras me hicieron ser lo que ahora te enorgullece, trataré de ser siempre el mejor hermano para ti, te amo Jessi.*

*A mis abuelos:*

*Que son mi origen divino, aún cuando nunca los disfrute como hubiera deseado se que sonrían junto conmigo porque mis éxitos son también los suyos, a mi abuelita Lilia, mi abuelita Flor, mis abuelos Daniel y Gabriel que seguramente festejan este momento único en mi vida, les dedico este proyecto, se que están orgullosos y yo lo estoy de ellos, los llevaré conmigo en cada paso que de, vivo con la esperanza de volver a verlos algún día, para así expresarles todo lo que me hacen falta y lo mucho que los amo, que Dios los bendiga y los cuide donde quiera que estén, los extraño mucho.*

*A ti amor:*

*Por ser mi gran amiga, por ser incondicional, por ser auténtica, por amarme inmerecidamente de esa forma que solo tu conoces, por soportar mis malos ratos y mantener la confianza en mi, por tus locuras, por tus lecciones de vida, por mostrarme los alcances de la felicidad, aunque pasen los años nunca dejaré de agradecer tu apoyo moral, tu tiempo y espacio, vivirás en mi como el mejor recuerdo de juventud, te adoro por ser una mujer valiente y darte por completo sin esperar nada a cambio, por respetarme y aceptarme tal como soy, gracias por dar lo mejor de ti y compartirlo conmigo, gracias Elizabeth Vizcarra Guillen, te amo.*

*A mis amigos:*

*Dicen por ahí que los amigos son aquellos hermanos que elegimos a lo largo de la vida, y es verdad, porque aún cuando han sido pocos, puedo decir que han sido buenos y leales, me apoyaron cuando mas los necesite y sin pedir nada a cambio me brindaron su confianza, su alegría y muchas veces su tristeza, han sido y serán por siempre una parte importante de mi historia, para mis amigos Ulises Arredondo, Rodolfo Trejo, Erick Zavala a los cuales conozco desde mi infancia y en lo especial a Rosario Palomino y Alejandra Moya, les dedico este trabajo para que nunca teman enfrentarse a la vida y jamás olviden uno de los tesoros mas hermosos que una persona puede tener, la lealtad de un verdadero amigo, gracias por caminar conmigo, los quiero mucho.*

*A mis profesores:*

*Ahora que veo el camino recorrido es cuando comprendo y valoro más sus palabras, cada uno de ustedes dejó en mi una gran enseñanza, que algunas veces no entendí pero con el paso del tiempo comprendí perfectamente, valoro su gran esfuerzo y dedicación a esta labor que aunque no los hace millonarios los hace merecedores del reconocimiento perdurable de los que ahora somos gentes de bien y comprometidos con nuestro país, gracias por su amor a la docencia y soportarnos tantos años, sin su magnífica labor este sueño jamás se hubiera convertido en realidad, que Dios les conserve ese corazón tan fuerte y humilde para seguir aportando a sus alumnos lo mejor de sus años y experiencia en este largo y sinuoso camino llamado vida.*

*A la profesora Silvia Vega Muytoy:*

*Gracias por predicar con el ejemplo, por demostrarme que somos los únicos dueños de nuestros éxitos o fracasos, por aceptar este proyecto y apoyarnos para darle forma, mil gracias por haber creído en nosotros, por aguantarnos tanto tiempo, por cada uno de sus valiosos consejos, le ofrezco en reconocimiento mi entero respeto porque usted sin duda alguna es una de las mejores universitarias que ha tenido México, personas como usted siempre harán falta en el mundo.*

*A la UNAM:*

*Gracias a la máxima casa de estudios de México por abrirme sus puertas y haberme permitido disfrutar de toda su belleza, su cultura y su historia como ninguna; forje dentro de ella los valores más significativos de mi vida y que hoy me hacen ostentar con gran orgullo mi origen universitario, no pude haber llegado a un lugar mejor para darme cuenta que la universidad es mucho más que un nombre, es en sí misma el verbo de la verdad que nos hace libres, por eso hoy vivo convencido que “por mi raza, hablará el espíritu”.*

*A la Facultad de Aragón:*

*Eternamente agradecido por haberme acogido los cinco años más importantes de mi vida profesional, por ser el lugar donde encontré la oportunidad de ser alguien mejor; como muchos llegue como un alumno más, ahora antes de partir devuelvo en este proyecto tan solo un poco de lo mucho que me dio, aquí conocí la competencia que me permitió madurar y crecer, comprendí que siempre se puede ser mejor de lo que uno mismo cree, que el buscar la verdad no es solo una expresión ensayada sino el modo más decente de vivir, conocí grandes compañeros que compartieron junto conmigo su tiempo así como sus conocimientos, así que prometo jamás olvidar esta hermosa casa de estudios que hizo realidad un sueño de noches enteras, me llevo contento todos los recuerdos, hoy dejo este humilde legado que es tan solo una forma de decir gracias y hasta siempre.*

*“La única persona con la que debes competir o a la que debes oponerte para conseguir el éxito es el hombre que te mira cuando estas frente al espejo”*

*“Para ser exitoso no tienes que hacer cosas extraordinarias.*

*Haz cosas ordinarias, extraordinariamente bien”*

*Muchas gracias a todos...*

*Con amor Alex,*

# *Agradecimientos de Martín*

---

*A Dios:*

*Por ser la luz que me ha guiado y que me ha permitido rendir este pequeño tributo en vida a mi madre.*

*A mi Madre: Guadalupe Alemán Cruz.*

*Sabiendo que no existe una forma de agradecerte toda una vida de sacrificios y esfuerzos, quiero que sientas que el objetivo logrado también es tuyo, y que la fuerza que me ayudó a conseguirlo es y ha sido tu amor incondicional.*

*A mis Hermanos: Leo, Ale, Toño, Beto y Bety.*

*Gracias por todo el apoyo que me han brindado en el transcurso de mi vida, por toda la ayuda recibida, ya que han hecho más ligero mi camino. Quiero agradecer todo el amor, paciencia y comprensión para conmigo.*

*A mis seres más queridos y Amigos.*

*Gracias.*

# Índice

INTRODUCCIÓN	v
OBJETIVO	vii
JUSTIFICACIÓN	viii
<b>CAPÍTULO 1. MARCO TEÓRICO Y ANTECEDENTES DE LA CALIDAD DEL SOFTWARE</b>	
Marco Histórico	9
1.1 Concepto de Calidad	11
1.2 Definiciones sobre la Calidad	13
1.2.1 Definiciones Relacionadas con el Cliente	14
1.3 Precusores de la Calidad Mundial	15
1.3.1 Philip B. Crosby	15
1.3.2 Joseph M. Juran	17
1.3.3 Edwards W. Deming	19
1.3.4 Kaoru Ishikawa	21
1.4 La Calidad en México	23
1.5 La Calidad relacionada al Desarrollo de Sistemas	27
1.5.1 La Industria del Software en México	29
1.5.2 Problemática del software en el México actual	31
1.6 El Modelo de Calidad como Estrategia de Solución	34
<b>CAPÍTULO 2. DEFINICIÓN DEL MODELO CMMI</b>	
Objetivo	36
Justificación	37
2.1 Modelos y estándares de calidad	38
2.2 Metodología de desarrollo de software	41
2.2.1 Importancia en el uso de una metodología específica	42
2.3 Mejoramiento de procesos	44
2.3.1 ¿Qué significa mejorar el proceso?	45
2.4 Modelo CMMI (Capability Maturity Model Integration)	47
2.4.1 Breve Historia de Desarrollo	47
2.5 Seleccionando un modelo CMMI	48
2.5.1 Systems Engineering (Ingeniería de Sistemas)	49
2.5.2 Software Engineering (Ingeniería de Software)	49
2.5.3 Integrated Product and Process Development (Desarrollo integrado de productos y procesos)	49

2.5.4 Supplier Sourcing (Proveedor de Recursos)	50
2.6 Usos sugeridos de este modelo	50
2.7 Componentes del modelo CMMI	52
2.7.1 Niveles de madurez	53
2.7.1.1 Nivel 1 o Inicial	54
2.7.1.2 Nivel 2 o Repetible	55
2.7.1.3 Nivel 3 o Definido	57
2.7.1.4 Nivel 4 o Administrado	59
2.7.1.5 Nivel 5 u Optimizado	61
2.7.2 Áreas del proceso	63
2.7.3 Metas específicas	63
2.7.4 Prácticas específicas	63
2.7.5 Metas genéricas	63
2.7.6 Prácticas genéricas	64
2.7.7 Características de la Institucionalización	64
2.8 Características comunes	65
2.9 Prácticas genéricas por característica común	66
2.9.1 Compromiso para realizar (práctica genérica)	66
2.9.1.1 Establecer una política organizacional	66
2.9.2 La habilidad para realizar (práctica genérica)	67
2.9.2.1 Planear el proceso	67
2.9.2.2 Proveer los recursos	67
2.9.2.3 Asignar responsabilidades	68
2.9.2.4 Capacitar al personal	68
2.9.2.5 Establecer un proceso definido	69
2.9.3 Dirigiendo la aplicación (práctica genérica)	70
2.9.3.1 Identificar e involucrar a las personas clave	70
2.9.3.2 Monitorear y controlar los procesos	70
2.9.3.3 Recopilar información para el mejoramiento de los procesos	71
2.9.4 Verificando la aplicación (práctica genérica)	71
2.9.4.1 Evaluación objetiva	71
2.9.4.2 Análisis del estado del proceso en la alta gerencia	71
2.10 Características de un proceso maduro	72

<b>CAPÍTULO 3. APLICACIÓN DEL MODELO CMMI</b>	
<b>Objetivo</b>	<b>75</b>
<b>3.1 Modelo Metodológico de Desarrollo de Soluciones</b>	<b>76</b>
<b>3.2 Proceso de Administración de Proyectos</b>	<b>81</b>
3.2.1 ¿Qué es un Proyecto?	81
3.2.2 Gestión de la Calidad	83
3.2.3 Riesgos	83
3.2.4 Comunicación	84
3.2.5 Proceso de inicio	85
3.2.6 Planeación y Estimación	87
3.2.7 Ejecución y seguimiento del plan	90
3.2.8 Control y métricas	92
3.2.9 Administración de desviaciones	96
3.2.10 Control de cambios	98
3.2.11 Evaluación y cierre	100

<b>CAPÍTULO 4. PROPUESTA DE UNA ESTRATEGIA DE CALIDAD</b>	
<b>Objetivo</b>	<b>103</b>
<b>4.1 Descripción genérica del CDR (Centro Desarrollo Regional)</b>	<b>104</b>
4.1.1 Funciones principales	104
4.1.2 Estándares guía del área de diseño	106
4.1.2.1 Diseño funcional	108
4.1.2.2 Desarrollo del diseño	109
4.1.2.3 Pruebas de aceptación	109
4.1.2.4 Concepto de oficina	110
4.1.2.5 Consideraciones al diseñar	110
4.1.3 Estándares CDR	113
4.1.3.1 Definición de programas	114
4.1.3.2 Paquetización y paralelismo	116
4.1.3.3 Control de versiones	118
4.1.3.4 Control de versiones de componentes	119
4.1.3.5 Nomenclatura de componentes	120
4.1.4 Estándares guía para el área de fábrica de software	121
4.1.4.1 Estructuras lógicas	122
4.1.4.2 Estándares de acceso a base de datos	123

4.1.5 Estándares guía para el área de pruebas CDR	124
4.1.5.1 Definición de la estrategia del proyecto (diseño, construcción, pruebas)	125
4.1.5.2 Revisar y complementar matrices de pruebas	125
4.1.5.3 Preparar entorno y datos de pruebas	127
4.1.5.4 Ejecución de pruebas	128
4.1.5.5 Administrar el proceso de pruebas	128
4.1.5.6 Construcción de entornos	130
4.1.5.7 Pruebas CDR (Centro Desarrollo Regional)	130
4.1.6 Relación del Centro Corporativo Regional con el CDR	132
CONCLUSIONES	137
GLOSARIO	138
BIBLIOGRAFÍA	139

# *Introducción*

---

---

A lo largo de su historia, el hombre se ha enfrentado a un sin fin de adversidades usando solo el sentido común, siendo esta la forma mas simple y natural, muchas veces funciona pero no siempre resulta así, no todo puede solucionarse sin contar con una estrategia bien definida y mucho menos cuando aquello que se desea realizar o resolver conlleva gran complejidad. Como sabemos, hacer las cosas de esta manera un tanto incorrecta solo ha llevado a cometer un sin fin de errores, que en si mismos podrían considerarse no del todo malos, los verdaderos problemas se presentan cuando se repiten con frecuencia dando lugar a un ciclo vicioso del cual seguramente será difícil escapar.

Si bien el ser humano se ha preocupado por hacer las cosas de la mejor forma posible, no deberíamos omitir entonces la calidad inherente como la manera más segura de lograr el objetivo final que es hacer las cosas bien desde un principio y con esto evitar el desperdicio de recursos lo cual origina improductividad. En la actualidad, todas aquellas empresas que emplean buenas estrategias de calidad marcan el camino a seguir y llevan la delantera que los mantiene compitiendo en los mercados mundiales más importantes.

Afortunadamente siempre ha existido gente comprometida con la calidad en los productos y servicios ofrecidos, personajes de la historia que han sido la punta de lanza de lo que hoy conocemos como calidad total, algunos tan conocidos como Philip B. Crosby, Joseph M. Juran, Edwards W. Deming y el profesor Kaoru Ishikawa solo por mencionar algunos, aportaron a la humanidad grandes enseñanzas que hoy día se traducen en adelantos tecnológicos, en mejores productos y por ende en una mejor calidad de vida.

En México la calidad no ha sido precisamente un tema prioritario en el gobierno y en las instituciones que deberían respaldarla, en cambio en algunos otros países tales como Japón, Francia, Inglaterra, Alemania e India solo por citar algunos, ha sido un estilo de vida que adoptaron desde hace años y que les ha otorgado todos los beneficios de los que hoy disfrutan.

Dado que la calidad ha demostrado ser la forma más óptima de realizar proyectos que permiten alcanzar los mayores estándares de productividad, se desarrollo un modelo de madurez de capacidades en la Universidad de Pittsburg en los Estados Unidos de Norteamérica, el cual actualmente goza de gran reconocimiento y valor entre las empresas e instituciones de mayor importancia en el mundo. Dicho modelo contempla una serie de prácticas a realizar por el personal de cada empresa para conseguir los objetivos planteados minimizando al máximo los errores tradicionales que muchas veces merman la calidad de lo que se produce.

Todo modelo en si mismo es bueno, pero el éxito que tenga al ser implementado depende en gran medida de la misma calidad de la gente, desde ahí comienza todo y este proyecto esta dirigido a desarrollar precisamente una estrategia que parte del uso de las buenas prácticas del modelo CMMI (Capability Maturity Model Integration por sus siglas en inglés) y que permiten a muchas empresas alcanzar niveles de productividad mucho mas óptimos, con esto se logra competir en los grandes mercados internacionales y al final se traduce en mejores productos así como en servicios más adecuados para la población que demanda cada vez una mayor calidad en todo lo que se produce.

Por todos es sabido que las instituciones financieras ofrecen una gran cantidad de productos y servicios de vital importancia en el mundo actual, en el afán de capturar la mayor cartera de clientes dichas empresas recurren a grandes planes comerciales que deben su éxito, en gran medida, en el respaldo que solo la calidad puede darles, es por ello que se desarrolla una estrategia que permite crear sistemas financieros que den soluciones a todas las demandas de crecimiento en forma óptima, para ello, el enfoque principal se encuentra en la idiosincrasia de México, aprovechando la gran capacidad que posee la gente y en virtud de algún día eliminar la dependencia tecnológica con otros países y ser en si mismos un ejemplo en esta y en las futuras generaciones.

# *Objetivo*

---

---

A lo largo de los últimos decenios la economía mundial ha transitado un proceso de diversificación industrial derivado del auge de las nuevas tecnologías de la información, las cuales han cambiado la organización de la industria e incluso las formas de convivencia y la vida misma. Una de las industrias con mayor peso en esta dinámica es la del software, que materializa la operación de todas las innovaciones y constituye hoy por hoy el corazón de la nueva economía.

Por ello, algunos países en desarrollo buscan participar de esta nueva ola de avance tecnológico y lo están logrando con éxito, como la India, en particular la región sureña de Bangalore, donde opera un enclave de desarrollo tecnológico basado en la prestación de servicios de software para el resto del mundo. De igual modo, Israel e Irlanda han logrado abrirse paso en el mercado para consolidar su propia industria del software.

Ante el éxito de estos países, otros buscan alentar su propia industria, entre ellos China y Brasil; si bien puede ser en efecto una oportunidad, la experiencia internacional muestra que para que estos esfuerzos cristalicen se requiere la participación del gobierno, los productores de software y los centros de desarrollo universitarios.

En este sentido el desarrollo de una industria de software dinámica en México no debe imitar modelos ajenos, sino elaborar uno propio que aproveche las ventajas de la diversidad de la cultura mexicana.

El propósito de este trabajo es delinear un enfoque que rescate esta habilidad para encontrar un punto de entrada adecuado a la globalización de la industria del software, en vez de sólo intentar retomar modelos externos. El éxito de la experiencia mexicana dependerá de su habilidad para formular una estrategia particular de inserción en la globalidad.

# *Justificación*

---

---

El software es una industria muy atractiva, ya que es intensiva en conocimiento y en mano de obra calificada. Sin embargo, requiere una fuerte disciplina de los países para contar con la fuerza laboral que exige la industria, por lo que para las economías emergentes como México el reto es ajustar la capacitación con el desarrollo de la industria.

Los escépticos argumentan que las tecnologías de la información (TI) son la última moda y que, por ende, no hay que dejarse llevar por el canto de las sirenas. Sin embargo, los líderes del mercado ponen como ejemplo los casos exitosos de la India, Israel e Irlanda, que han traducido su creciente éxito en las tecnologías de la información – en especial las exportaciones de software – en generación de empleo, ganancias, divisas y crecimiento económico.

En este sentido, se puede argumentar que los modelos de innovación son los que pueden permitir que la industria se especialice en servicios de software con una alta competencia con otros países, por lo que es necesario pensar que lo relevante es el diseño más que las tareas repetitivas de eliminación de errores de programación.

Para que la industria del software crezca necesita mano de obra calificada y programas de apoyo por parte del gobierno, tal como se ha hecho en todos los países que han logrado el éxito. Esto es indispensable en el ámbito mexicano si se quiere dar certidumbre al desarrollo de la industria del software en el mediano plazo.

La justificación de este trabajo es la necesidad de consolidar la industria de servicios de software para satisfacer el mercado interno y que un segmento de esta industria continúe obteniendo provecho de su cercanía con Estados Unidos, y en esta perspectiva se debería fortalecer el nicho de mercado de los servicios de software.

# Capítulo 1



## Marco Histórico

La historia de la humanidad está directamente ligada con la calidad desde los tiempos más remotos, el hombre al construir sus armas, elaborar sus alimentos y fabricar su vestido observa las características del producto y enseguida procura mejorarlo.

La práctica de la verificación de la calidad se remonta a épocas anteriores al nacimiento de Cristo. En el año 2150 A.C., la calidad en la construcción de casas estaba regida por el **Código de Hammurabi**, cuya regla No. 229 establecía que *"si un constructor construye una casa y no lo hace con buena resistencia, si la casa se derrumba y mata a los ocupantes, el constructor debe ser ejecutado"*<sup>1</sup>. Los fenicios también utilizaban un programa de acción correctiva para asegurar la calidad, con el objeto de eliminar la repetición de errores. Los inspectores simplemente cortaban la mano de la persona responsable de la calidad insatisfactoria.

En los vestigios de las antiguas culturas también se hace presente la calidad, ejemplo de ello son las pirámides Egipcias, los frisos de los templos griegos, etc. Sin embargo, la **Calidad Total**, como concepto, tuvo su origen en Japón donde ahora es una especie de religión que todos quieren practicar.

Durante la **edad media** surgen mercados con base en el prestigio de la calidad de los productos, se popularizó la costumbre de ponerles marca y con esta práctica se desarrolló el interés de mantener una buena reputación (las sedas de damasco, la porcelana china, entre otras). Dado lo artesanal del proceso, la inspección del producto terminado es responsabilidad del productor que es el mismo artesano.

---

<sup>1</sup> The Code of Hammurabi, (<http://www.wsu.edu/~dee/MESO/CODE.HTM>).

Con el advenimiento de la **era industrial** esta situación cambió, el taller cedió su lugar a la fábrica de producción masiva, bien fuera de artículos terminados o bien de piezas que iban a ser ensambladas en una etapa posterior de producción.

La era de la revolución industrial, trajo consigo el sistema de fábricas para el trabajo en serie y la especialización del trabajo. Como consecuencia de un alta demanda aparejada con el espíritu de mejorar la calidad de los procesos, la función de inspección llega a formar parte vital del proceso productivo y es realizada por el mismo operario (el objeto de la inspección simplemente señalaba los productos que no se ajustaban a los estándares deseados).

**A fines del siglo XIX** y durante las tres primeras décadas del **siglo XX** el objetivo es producción. Con las aportaciones de Taylor<sup>2</sup> la función de inspección se separa de la producción; los productos se caracterizan por sus partes o componentes intercambiables, el mercado se vuelve más exigente y todo converge a producir.

El cambio en el proceso de producción trajo consigo cambios en la organización de la empresa. Como ya no era el caso de un operario que se dedicara a la elaboración de un artículo, fue necesario introducir en las fábricas procedimientos específicos para atender la calidad de los productos fabricados en forma masiva. Dichos procedimientos han ido evolucionando, sobretudo durante los últimos tiempos.

El control de la calidad se practica desde hace muchos años en Estados Unidos y en otros países, pero los japoneses, enfrentados a la falta de recursos naturales y dependientes en alto grado de sus exportaciones para obtener divisas que les permitieran comprar en el exterior lo que no podían producir internamente, se dieron cuenta de que para sobrevivir en un mundo cada vez más agresivo comercialmente, tenían que producir y vender mejores productos que sus competidores internacionales como Estados Unidos, Inglaterra, Francia y Alemania.

Lo anterior los llevó a perfeccionar el concepto de calidad. Para ellos debería haber calidad desde el diseño hasta la entrega del producto al consumidor, pasando por todas las acciones, no sólo las que incluyen el proceso de manufactura del producto, sino también las actividades administrativas y comerciales, en especial las que tienen que ver con el ciclo de atención al cliente incluyendo todo servicio posterior.

---

<sup>2</sup> Frederick W. Taylor, hizo posible una gran expansión de la manufactura y el consumismo, naciendo así grandes empresas como la de Henry Ford.

## 1.1 Concepto de Calidad

Del latín “*Qualitatem*” y cuyo significado es *atributo o propiedad que distingue a las personas, a bienes y servicios. Esta distinción implica nivel de excelencia, pero algo excelente no es algo caro y lujoso, sino adecuado para su uso*<sup>3</sup>. En su acepción latina Calidad (*Qualitas*) significa forma o manera de ser. Esto significa que los servicios y los productos llevarán siempre el sello de la personalidad de quienes los elaboran, lo que en otras palabras quiere decir que la calidad sólo podrán ofrecerla personas de calidad.

Esto lleva implícitos conceptos como educación, cultura, valores personales y actitud, entre otros. En el transcurso del tiempo esta palabra ha tenido múltiples definiciones que expresan más o menos lo mismo: la satisfacción de expectativas y el cumplimiento o superación de los requerimientos de los clientes.

Sin embargo, el verdadero significado no está en la palabra en sí, sino en lo que hay alrededor de ella. La calidad productiva está íntimamente ligada a la calidad de vida. Se debe recordar que tanto empleados (productores) como clientes (consumidores) son ante todo seres humanos. Hay que mencionar también que la calidad no es un fin en sí misma, sino un medio para alcanzar fines de mayor trascendencia: desarrollo humano, una mejor distribución de la riqueza y una mejora general del bienestar.

Calidad es precisión, desarrollo humano, uso de tecnologías duras y suaves, alternas y avanzadas, orgullo por lo bien hecho, trabajo en equipo, autoconfianza, autocontrol, autoplaneación, es una actitud de superación constante, de perfeccionamiento continuo, de búsqueda comprometida para lograr siempre lo mejor; no conformarse; es tener un compromiso con uno mismo, con la organización en la que se trabaja y con la comunidad en la que se vive.

Por eso la calidad no es simplemente una metodología o un conjunto de metodologías a implantar dentro de una organización, sino un valor o un conjunto de valores que generan actividades y comportamientos en el trabajo y fuera de éste. Es alcanzar los máximos estándares en todo lo que realizamos. Es una filosofía y un estilo de vida.

---

<sup>3</sup> “Calidad Estratégica Total”, Rubén Roberto Rico, 1993. Páginas 4 y 5.

La palabra calidad designa el conjunto de atributos o propiedades de un objeto que permite emitir un juicio de valor acerca de él; en este sentido se habla de la nula, poca, buena o excelente calidad de un objeto.

Cuando se dice que algo tiene calidad, se designa un juicio positivo con respecto a las características del objeto, el significado del vocablo calidad en este caso pasa a ser equivalente al significado de los términos excelencia o perfección.

El concepto de perfección durante la Edad Media era tal, que se consideraba como obra perfecta sólo aquella que no tenía ningún defecto. La presencia de uno de estos por pequeño que fuera, era suficiente para calificar a la obra como imperfecta.

Los trabajos de manufactura en la época preindustrial, como eran prácticamente labores de artesanía, tenían mucho que ver con la obra de arte, el artesano ponía todo su empeño en hacer lo mejor posible cada uno de sus trabajos cuidando incluso que la presentación cumpliera los gustos estéticos de la época, dado que de la perfección de su obra dependía su prestigio artesanal.

El juicio acerca de la calidad del producto tenía entonces como base la relación personal que se establecía entre el artesano y el usuario. Cuando alguien necesitaba de un producto, como podría ser una herramienta o un determinado vestido o traje, exponía sus necesidades al fabricante, quien lo elaboraba de acuerdo con los requerimientos establecidos por el cliente. Como eran trabajos hechos a la medida, el productor sabía de inmediato si su trabajo dejaba satisfecho al cliente o no.

## 1.2 Definiciones sobre la Calidad

Existe una gran variedad de definiciones actuales sobre “calidad”, muchas de ellas parecen ser demasiado rebuscadas y otras definen este concepto de forma muy concreta. Lo cierto es que todas ellas mantienen un factor común. La mayor parte de las definiciones de calidad se refieren a “adecuado para su uso” o “conformación a los requerimientos”.

Existen varias versiones oficiales de calidad, que han sido preparadas por diferentes instituciones, tales como la British Standards Institution (BSI), la American Society for Quality Control (ASQC), la European Organization for Quality Control (EOQC) y la International Organization for Standardization (ISO), entre otras.

Precisamente de lo anterior se desprenden las siguientes definiciones que son un compendio desde diferentes puntos de vista:

- a) *Definición trascendente.* La calidad no es ni espíritu ni materia, sino una tercera entidad independiente de ambas. Aunque la calidad no puede definirse todo mundo sabe que es.
- b) *Definición basada en el producto.* Las diferencias de calidad representan diferencias en la cantidad de alguno de los ingredientes y atributos deseados.
- c) *Definición basada en el usuario.* La calidad consiste en la capacidad para satisfacer las expectativas.
- d) *Definición basada en la fabricación.* La calidad significa la conformidad del producto con los requerimientos.
- e) *Definición basada en su valor.* La calidad es el grado de excelencia con un precio aceptable y un control de variabilidad a un costo aceptable.

### **1.2.1 Definiciones Relacionadas con el Cliente**

Al hablar de calidad, de manera paralela, se piensa en la relación producto-cliente, para este caso el producto sería el software y el cliente sería la institución financiera. Obviamente la calidad no es exclusiva del software o su desarrollo, es tan importante para ingenierías como para todas las áreas ya que en todos los casos siempre esta involucrado el factor satisfacción para el cliente.

Por esto mismo se mencionan a continuación seis definiciones que se consideran relevantes y absolutamente relacionadas a la comprensión de la calidad como un medio estratégico para cumplir y sobrepasar las expectativas del cliente:

- a) La calidad es el atributo clave que los clientes utilizan para evaluar los productos y servicios.
- b) La calidad involucra a todas las áreas de una empresa, cualquiera que ésta sea, para satisfacer los requerimientos totales de todos los clientes, sin importar quién sea éste.
- c) La calidad esta dictada por el mercado, la competencia y especialmente por el cliente.
- d) El concepto moderno de calidad rechaza la noción tradicional de que la calidad es el grado de conformidad con un estándar o la medición de la bondad de un trabajo. El concepto japonés de calidad gira alrededor de lo “adecuado de un producto” y el grado de satisfacción que el cliente obtiene con el uso del producto. No son los fabricantes sino los clientes los que deciden si se ha logrado obtener un producto de calidad.
- e) La calidad consiste en cumplir y exceder las expectativas del cliente para preservar el futuro del negocio.
- f) La calidad es la capacidad de un producto o servicio para satisfacer a conciencia la combinación preconcebida de deseos de los clientes, que están claramente relacionados con las características de desempeño o apariencia, y que no causen reacciones evidentes u ocultas en otras personas.

## **1.3 Precursores de la Calidad Mundial**

Dentro de este contexto histórico es imposible dejar de mencionar a cuatro grandes personajes del siglo XX y que aún en nuestros días siguen influenciando de manera sobresaliente en el amplio mundo de la calidad, ellos conformaron lo que hoy día se conoce y se practica como *calidad total*; se presentan a continuación aspectos de su vida y obra, así como su aportación al mundo de manera sencilla sin omitir lo esencial de cada uno para entender lo que ha llevado a lograr la excelencia en los productos y servicios que hoy en día tanto se disfrutan.

El desarrollo de software de calidad encuentra sus inicios en las enseñanzas y aportaciones que dejaron huella al paso de los años, tocando a todas las áreas del conocimiento destacando las que tienen que ver actualmente con el desarrollo de sistemas.

### **1.3.1 Philip B. Crosby**



Nació en Wheeling, West Virginia el 18 de junio de 1926. Empezó trabajando como un profesional de calidad en 1952 después de servir en la Segunda Guerra Mundial y Corea, y al mismo tiempo sirviendo a una escuela médica.

La carrera de Crosby empezó cuando decidió que su meta sería enseñar dirección y difundió la idea de la prevención de problemas en vez de corregirlos.

Con 40 años como filósofo en los negocios, Philip B. Crosby publica "Hands on" sobre su experiencia en la dirección de empresas. Las conferencias de Crosby proporcionan una discusión pensativa y estimulante del papel de la dirección para encaminar a sus organizaciones, sus empleados, sus proveedores, y de ellos mismos para tener éxito. Crosby usaba historias de vidas reales junto con anécdotas aplicables para dar a entender sus ideas.

En 1979 fundó "Asociados Philip Crosby", Inc. (PCA, Philip Crosby Associated por sus siglas en inglés), y durante los próximos diez años creció aún más en una organización comercializadora con 300 empleados alrededor del mundo y \$80 millones de dólares en réditos. PCA enseñó a la alta gerencia cómo establecer una cultura preventiva para producir cosas de buena calidad desde la primera vez.

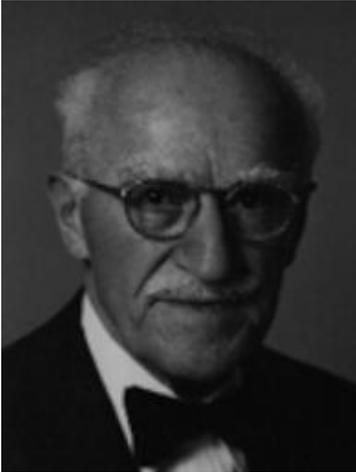
General Motors, Chrysler, Motorola, Xerox, hospitales, y cientos de corporaciones mundiales utilizan las ideas de PCA para entender y controlar la dirección durante un proceso de calidad. Hasta la fecha PCA todavía enseña sus técnicas en 16 idiomas alrededor del mundo. En 1991 Crosby se retiró de PCA y fundó "Career IV", Inc., una compañía que proporcionó conferencias y seminarios dirigidos a ayudar a los ejecutivos con perspectivas de crecimiento y actualización. En 1997 compro acciones de PCA y estableció "Philip Crosby Associates II", Inc., actualmente opera en más de 20 países alrededor del mundo.

Crosby confirma que la calidad está basada en cuatro principios absolutos:

- 1) ***La calidad implica cumplir con los requerimientos.*** La fijación de los requerimientos es una de las responsabilidades de la gerencia, como también lo son sistemas de comunicación y su efectividad. Afirma que, si los directivos quieren que el personal "realice las cosas bien desde la primera vez", deben informar claramente a todo el mundo en que consiste hacer las cosas bien, para cada empresa las condiciones pueden variar.
- 2) ***La calidad proviene de la prevención.*** El primer principio absoluto fue la comprensión del proceso por el cual se deben prever claramente aquellas circunstancias que pueden provocar errores en la producción de bienes y servicios. El segundo se refiere a identificar y eliminar todos los riesgos que no deben tomarse innecesariamente.
- 3) ***El estándar de calidad es cero defectos.*** Se trata de que el producto / servicio se ajuste a los requerimientos y, de acuerdo con Crosby, éste debe ser también el estándar de desempeño personal de todos los miembros de la organización, que proviene de un cambio de actitud.
- 4) ***La medición de la calidad es el precio de la inconformidad.*** De acuerdo con Crosby las compañías manufactureras gastan el 25% de sus ventas haciendo las cosas mal y las compañías de servicio consumen el 40% de sus costos de operación en acciones igualmente desperdiciadas.

***Philip B. Crosby*** sustenta el siguiente concepto de calidad. "*La idea esencial del movimiento de calidad es la prevención, la calidad es gratis, sus costos sólo están relacionados con los diversos obstáculos que impiden que los operarios la obtengan desde la primera vez*".

### 1.3.2 Joseph M. Juran



Joseph Moses Juran nació el 24 de diciembre de 1904 en la ciudad de Braila, entonces parte del imperio Austrohúngaro, ahora es parte de Rumania. Su padre, Jacob, fue zapatero del pueblo en Rumania, pero tuvo problemas para encontrar trabajo en América donde el calzado se hacía en fábricas. Juran decía, "Mis padres no tenían ningún problema de calidad. Nunca tenían un fracaso de poder, nunca tenían un problema automovilístico. Por supuesto, ellos no tenían poder; ellos no tenían ningún automóvil".

Juran ha sido activo durante casi todo el siglo XX. Él se ha levantado de la pobreza para hacerse de una vida prospera muy variada, como escritor, educador y consultor.

La mayor contribución de Juran al mundo ha estado en el campo de dirección, particularmente la dirección de calidad. Observador astuto, oyente atento, sintetizador inteligente y pronosticador del presente, se ha llamado a si mismo el "padre" de la calidad, el "guru" de la calidad y el hombre que "les enseñó calidad a los japoneses". Él se reconoce como la persona que dio origen a la calidad y el concepto de lo que hoy se conoce como "Dirección de Calidad Total".

En 1937, conceptuó el principio de Pareto en el que millones de gerentes confían para ayudar a separar las pocas cosas vitales de las muchas cosas útiles en sus actividades. Escribió el trabajo de referencia de estándares en el control de calidad, el "Manual de Mando de Calidad", primero publicado en 1951 y ahora en su cuarta edición.

También trabajó como consultor de negocios y organizaciones en cuarenta países, y ha hecho muchas otras contribuciones a la literatura en más de veinte libros y una gran cantidad de artículos publicados (en un total de diecisiete idiomas) así como las docenas de programas de entrenamiento de video.

Steve Jobs, fundador de Macintosh se refiere con temor a la "contribución profunda" de Juran. El Sr. Jungi Noguchi, director ejecutivo de la Unión japonesa de Científicos e Ingenieros, declara categóricamente que el Dr. Juran es la más grande autoridad en control de calidad en el mundo entero.

El enfoque de Juran sobre la administración de la calidad se basa en lo que se llama la trilogía de Juran:

1) ***Planeación de la calidad.***

- ✓ Comprobación de las virtudes de dicho proceso.
- ✓ Desarrollo de las características del producto.
- ✓ Determinación de las necesidades reales.
- ✓ Establecimiento de las metas de calidad.
- ✓ Análisis y desarrollo de un proceso.
- ✓ Identificación de los clientes.

2) ***Control de calidad.***

- ✓ Interpretación de las diferencias (realidad contra estándar).
- ✓ Establecimiento de los estándares de desempeño.
- ✓ Selección de las unidades de medición.
- ✓ Selección de los objetivos de control.
- ✓ Medición del desempeño real.
- ✓ Corrección de las diferencias.
- ✓ Fijación de las mediciones.

3) ***Mejoras de calidad.***

- ✓ Organización para el diagnóstico-descubrimiento de las causas.
- ✓ Implantación de los controles para conservar lo ganado.
- ✓ Demostración de la necesidad de las mejoras.
- ✓ Identificación de los proyectos a realizar.
- ✓ Organización para dirigir los proyectos.
- ✓ Diagnóstico para determinar las causas.
- ✓ Definición de las correcciones.

*Joseph M. Juran* mencionaba que “La calidad esta basada en la plena identificación de los clientes y la determinación de sus necesidades, así como la prevención de problemas crónicos y control de procesos, verificando el desempeño real conjunto”.

### 1.3.3 Edwards W. Deming



Nació el 14 de octubre de 1900 en la ciudad de Sioux, en 1917, ingresó a la Universidad de Wyoming en Laramie. En 1921 se graduó en ingeniería eléctrica. En 1925, realizó una maestría en la Universidad de Colorado y en 1928 un doctorado en la Universidad de Yale. Ambos grados fueron en las matemáticas y en la física. Deming estudió también la teoría de la música, aprendió a tocar varios instrumentos y compuso dos temas, varios cánticos, entre otros.

Como buen estadista, tuvo la misión conjunta de observar las elecciones griegas, en enero y abril de 1946; entre julio y octubre de 1946 fue consultor en una prueba al gobierno de la India, esto fue en enero y febrero de 1947; en marzo de 1971 es delegado al Congreso de la ciencia de la India exactamente en Nueva Delhi. En 1947 y 1950 en Tokio, fue consejero de pruebas técnicas del comando supremo de las potencias aliadas, fue maestro y consultor de la industria japonesa, a través de la “Unión de Científicos e Ingenieros Japoneses” en los años 1950 a 1952, 1955, 1960 y 1965.

De 1947 a 1952 fue miembro de las Naciones Unidas en la sub-comisión de pruebas estadísticas. En 1954 y 1955, consultor del Censo de México, trabajo para el Banco de México, y para el Ministerio de Economía.

Deming fue el primer científico occidental en ser invitado por los japoneses para conducir una serie de seminarios para trabajadores y gerentes, con respecto al uso de gráficas de control y técnicas estadísticas, orientadas al control de calidad. Fomentó la idea de ir más allá de las estadísticas para luchar por mejoras continuas, usando lo que después se conoció como “ciclo de Deming” que consiste en planear, ejecutar, comprobar y actuar (PECA).

Deming señala que existen obstáculos que impiden llegar a la calidad total tal como él los llama, se mencionan a continuación los “pecados capitales” o “enfermedades mortales”:

- 1) *Falta de consistencia.* Falta de consistencia en los propósitos de permanencia del negocio, al no planear los productos y servicios del futuro, apuntando a mercados específicos para que la compañía progrese y sea una fuente de empleos.
- 2) *Utilidades a corto plazo.* Las estrategias a corto plazo derrotan a la constancia de propósitos para sobrevivir con crecimiento a largo plazo.
- 3) *Evaluaciones de desempeño.* Los efectos de las evaluaciones de desempeño son devastadores.
- 4) *Cambios de empleo.* La movilidad de los directivos causa inestabilidad, y conduce a la toma de decisiones por parte de personas que tienen pocos conocimientos y poca comprensión de las actividades empresariales.
- 5) *Uso exclusivo de cifras visibles.* Los directivos no deben fijarse únicamente en las cifras visibles. Aunque éstas son importantes, la gerencia debe aprender a manejar el negocio con un enfoque más amplio y global (las cifras desconocidas son también muy importantes).

*Edwards W. Deming* da su propia definición de calidad. “*La calidad debe iniciarse en la alta gerencia y que todo el personal de la organización participe, se debe permanecer en un proceso continuo de mejoras basadas en aspectos científicos para llegar al objetivo principal que es la de servir siempre mejor al cliente*”.

### **1.3.4 Kaoru Ishikawa**



El Dr. Kaoru Ishikawa nació en Japón en el año 1915 y falleció en 1989. Se graduó en el Departamento de Ingeniería de la Universidad de Tokio. Obtuvo el Doctorado en Ingeniería en dicha Universidad y fue promovido a profesor en 1960. Obtuvo el premio Deming y un reconocimiento de la Asociación Americana de la Calidad. Fue el primer autor que intentó destacar las diferencias entre los estilos de administración japonés y occidentales. Precursor de los conceptos sobre la calidad total en el Japón.

Tuvo una gran influencia en el resto del mundo, ya que fue el primero en resaltar las diferencias culturales entre las naciones como factor importante para el logro del éxito en calidad. Era un gran convencido en la importancia de la filosofía de los pueblos orientales.

Quizá la contribución más importante de Ishikawa ha sido su importante papel en el desarrollo de una estrategia de calidad específicamente japonesa. El sello de "hecho en Japón" es gran sinónimo de calidad, no sólo cubre las expectativas de muchas organizaciones, también por el ciclo de vida del producto.

Siempre fue consciente de la importancia del apoyo en la dirección de una empresa, el apoyo es un elemento importante en Japón, toda la estrategia de calidad se basa mucho en eso. El trabajo de Ishikawa sobre la dirección de todo un conjunto de elementos para alcanzar la calidad cubrió décadas. En los años cincuenta y principios de los sesentas, él desarrolló cursos de control de calidad para los ejecutivos y gerentes de diversas áreas. También colaboró para impulsar la Conferencia Anual de Control de Calidad Anual para la Alta Dirección en 1963.

Fue fundador de la Union of Japanese Scientists and Engineers (UJSE), entidad que se preocupaba por promover la calidad dentro de Japón durante la época de la post-guerra.

Ishikawa esta considerado en su país como el principal precursor de la administración de la calidad total. Se inspiró en los trabajos de Deming y Juran.

Entre sus más grandes aportaciones destacan las siguientes:

- 1) **Círculos de control de calidad.** Fue el primero en introducir este concepto y en ponerlo en práctica con éxito.
- 2) **Diagrama de hueso de pescado o diagrama de Ishikawa.** Se usan actualmente en todo el mundo en las mejoras continuas, para representar los análisis de causa-efecto.

El diagrama de Ishikawa ayuda a graficar las causas del problema que se analice. Es llamado “Espina de Pescado” por la forma en la que se van colocando cada una de las causas o razones que a nuestro entender originan un problema (ver figura 1.1). Tiene la ventaja que permite visualizar de una manera muy rápida y clara, la relación que tiene cada una de las causas con las demás razones que inciden en el origen del problema. En algunas oportunidades son causas independientes y en otras, existe una íntima relación entre ellas, las que pueden estar actuando en cadena.

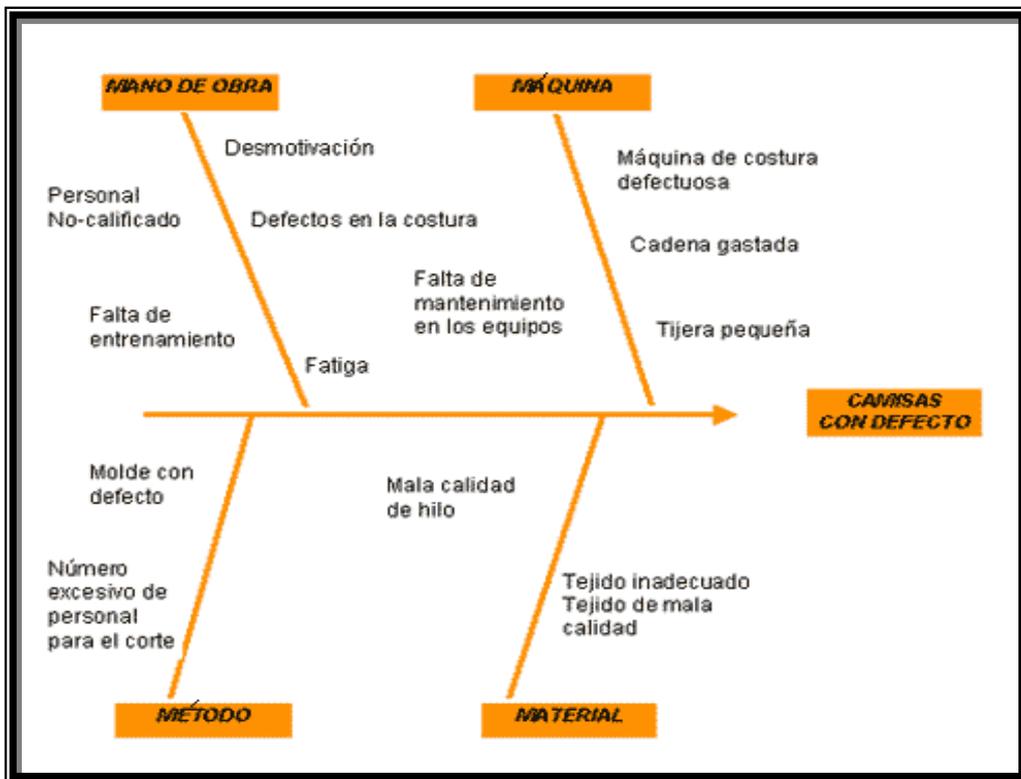


Figura 1.1 Ejemplo del Diagrama de Espina de Pescado

## **1.4 La Calidad en México**

En México aproximadamente el 40% de la población es menor de 19 años<sup>4</sup>. Es un país grande y joven con muchas necesidades insatisfechas y por lo tanto con muchas oportunidades.

El proceso de industrialización empezó en México a finales del siglo XIX, con la producción de textiles, alimentos, cerveza y tabaco. Sin embargo, no fue sino hasta 1940 cuando se inició un fuerte movimiento de industrialización basado en la política de sustitución de importaciones.

Dicha política adoptada por México ha generado entre otros, los siguientes problemas:

- a) El desarrollo de una economía que ofrece productos y servicios de baja calidad, poco competitivos en precio y calidad en los mercados internacionales.
  
- b) La sustitución de la capacidad instalada en las organizaciones principalmente industriales, sobre todo en lo que corresponde a la pequeña y mediana industria, que siempre se encuentra en una posición débil para afrontar problemas económicos como los ocurridos en 1976, 1986 y 1994.
  
- c) La actividad económica se concentra principalmente en 3 regiones: Ciudad de México, Guadalajara y Monterrey que acumulan el 70 % del valor agregado.
  
- d) México no ha sido capaz de desarrollar su propia tecnología, y depende constantemente de la importada con la consecuente baja competitividad.
  
- e) No hay suficiente ahorro e inversiones para renovar las instalaciones productivas y crear o mantener la infraestructura económica.

---

<sup>4</sup> Fuente: INEGI, consultado en septiembre de 2006.

Las empresas mexicanas crecieron acostumbradas a obtener ganancias fáciles y rápidas al poder operar protegidas de la competencia internacional, lo cual aunado al control de precios ejercido en los años setenta, reforzó su desinterés y apatía por invertir en nuevas tecnologías e infraestructura.

Por otro lado, conforme la economía fue progresando, la demanda de bienes intermedios y de capital se incrementó a grado tal que no pudo ser satisfecha por los proveedores nacionales.

Debido a la estrategia adoptada por México y a sus consecuencias relacionadas, el país no pudo compensar con exportaciones el incremento en importaciones, lo cual trajo como consecuencia un desequilibrio cada vez peor en relación con el comercio.

Para mediados de los ochenta se introdujeron políticas de liberalización comercial con el objeto de estimular las exportaciones. A partir de entonces, la economía mexicana ha experimentado una significativa transformación en respuesta a la amplia gama de reformas de la política comercial y de la búsqueda hacia los más altos estándares de calidad.

Sin embargo, es importante reconocer que los cambios hacia una economía abierta no pueden ser evaluados en un periodo corto, ya que necesitan de un plazo mayor para traer los beneficios esperados en términos de bienestar, debido a que otras variables no económicas como turbulencias políticas asociadas a la consolidación de la democracia, el mejoramiento en la calidad de la educación y los cambios en valores culturales (honestidad, puntualidad, disposición al trabajo y al ahorro, etc.) requieren quizá de 20 ó 30 años para madurar. Lo importante es que este proceso se ha iniciado.

Una característica de la economía mexicana es el alto grado de concentración, ya que las grandes corporaciones dominan sus respectivos mercados. La micro y pequeña empresa, que representan el 95% del total de empresas en México, son administradas en forma muy deficiente, operan por lo general con tecnología obsoleta y difícilmente pueden competir en el ámbito internacional.

En los últimos diez años, nuestro país se ha convertido en una nación estratégica para el resto del mundo, al ser el único que cuenta con tratados comerciales con los principales bloques económicos: TLC de América del Norte, TLC México Chile, TLC México Costa Rica, TLC México Nicaragua, TLC del Grupo de los Tres (Colombia, Venezuela y México), TLC México Unión Europea, TLC México Bolivia, TLC México Israel.

Muchos de los problemas de imagen de calidad de países como México han sido provocados por las políticas de economía cerrada y la consecuente falta de competencia a la que se enfrentan sus organizaciones. Una vez que la economía se abrió, el país se encontró de pronto en medio de un mundo altamente competido, y se vio forzado a mejorar rápidamente la calidad de sus productos y servicios.

La percepción de México como un país que sólo ofrece mano de obra barata cambia paulatinamente. Al día de hoy muchas de las corporaciones más importantes en el ámbito mundial incluyendo American Express, Ford Motor Company, Chrysler y General Motors, reportan que su fuerza laboral mexicana ofrece un nivel de calidad y confiabilidad no encontrada en otros países.

Las empresas nacionales también obtienen mejoras significativas en su desempeño, ofreciendo productos y servicios de mejor calidad. Aeroméxico ha logrado resultados tangibles a través de sus esfuerzos por incrementar la calidad de sus operaciones. Esta aerolínea reportó una puntualidad del 98.6 % en sus salidas, comparada con el 77.3 % que en promedio muestra Estados Unidos.

Para promover la competitividad en las organizaciones mexicanas, en 1989 se instituyó el **Premio Nacional de Calidad** en reconocimiento a las empresas que hayan logrado resultados sobresalientes en calidad, atención al cliente y calidad de vida en el trabajo. Las compañías que solicitan competir por el premio deben comparar en forma explícita su mejoramiento de calidad con respecto a otras compañías que participan en la misma categoría. Algunas de las empresas merecedoras al Premio Nacional de Calidad son, por ejemplo, General Motors planta Toluca, Crysel, Xerox México y American Express México que ha reducido hasta 76% su tiempo de respuesta de facturación.

Todas las plantas automotrices del país mantienen estándares de calidad mejores que el promedio mundial, además de que tienen niveles de productividad cercanos al promedio internacional, a pesar de la desventaja de contar con volúmenes de producción bajos. Un estudio de 1991 presentado por la Universidad de San Diego sobre Administración de Recursos Humanos, encontró que debido a las similitudes culturales entre México y Japón las maquiladoras japonesas habían podido implantar más fácilmente sus sistemas de calidad y así incrementar su productividad.

Las empresas mexicanas no podrán competir en los mercados internacionales si no mejoran en áreas como: calidad, productividad y distribución. Sólo aquellas empresas que estén certificadas podrán ofrecer sus productos o servicios a nivel global; debido a que en los últimos años la certificación de la calidad se ha convertido en la carta de presentación de las empresas para ingresar a nuevos mercados.

Las oportunidades a las que se enfrentan las empresas mexicanas las obliga a ser más competitivas en calidad, costo y servicios; dejar atrás la forma tradicional de administración y adoptar una nueva que les permita hacer frente a las presiones competitivas; y mejorar la calidad de todo el negocio, no sólo en las áreas de producción.

Hoy la calidad es buscada por muy pocas empresas mexicanas; la gran mayoría aún no han decidido involucrarse en un compromiso de esta naturaleza.

Pero una cultura de calidad en México está aún en proceso. La adopción de la nueva filosofía de calidad total requiere una transformación cultural, una nueva forma de administrar el negocio, donde el énfasis cambia de las utilidades a la satisfacción del cliente; en donde en lugar de administrar resultados se administran y mejoran continuamente los sistemas y procesos que los producen.

## **1.5 La Calidad relacionada al Desarrollo de Sistemas**

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto de software y la calidad del proceso de desarrollo de éste (calidad de diseño y fabricación). No obstante, las metas que se establezcan para la calidad del producto van a determinar los objetivos a establecerse durante el proceso de desarrollo. Sin estrategias adecuadas y perfectamente analizadas es prácticamente imposible obtener un buen producto. Este proceso constituye el objetivo fundamental del presente trabajo.

Pero la calidad del software se diferencia de la calidad de otro tipo de productos de fabricación industrial, ya que el software tiene características propias y muy específicas, tal como se menciona a continuación:

- a) El software es un producto mental, no restringido por las leyes de la Física o por los límites de los procesos de fabricación. Es algo abstracto, algo intangible.
  
- b) Se desarrolla, no se fabrica como un producto industrial común. El coste está fundamentalmente en el proceso de diseño, no en la posterior producción en serie, y los errores se introducen también en el diseño, no en la producción.
  
- c) Los costes del desarrollo de software se concentran en las tareas de ingeniería, mientras que en la fabricación clásica los costes se acentúan más en las tareas de producción.
  
- d) El software no se deteriora con el tiempo. No es susceptible de los efectos del entorno y su curva de fallos es muy diferente a la del hardware. Todos los problemas que surjan durante el mantenimiento estaban allí desde el principio y afectan a todas las copias del mismo; no se generan nuevos errores.

- e) Es artesanal en gran medida. El software, en su mayoría, se construye a medida, en vez de ser construido ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad.
  
- f) El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente del hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código máquina ejecutable.
  
- g) Es engañosamente fácil realizar cambios sobre un producto de software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.
  
- h) Como disciplina, el desarrollo de software es aún muy joven, por lo que las técnicas de las que dispone aún no están perfeccionadas.
  
- i) El software con errores no se rechaza. Se asume que es inevitable que el software presente algunos errores de poca importancia.

También es importante destacar que la calidad del software debe ser considerada en todos sus estados de evolución, ya sea en especificaciones, en el diseño, en los códigos, etc. No basta con verificar la calidad del producto una vez finalizado cuando los problemas de una mala estrategia de calidad ya dieron origen a situaciones que no tienen una solución adecuada o su reparación es muy costosa.

### **1.5.1 La Industria del Software en México**

Desde hace varios años, resulta evidente en México la necesidad de cambiar el paradigma con el que se ha construido la etapa de crecimiento y consolidación del estado moderno, cuya sentencia es: frente a la emergencia de las economías asiáticas, México no puede competir en base al precio de sus recursos humanos, ni debe interesarle mantenerlo deprimido; tampoco puede continuar apoyando sus finanzas internacionales en la exportación de materias primas con escaso o nulo valor agregado.

El paradigma de ser un país con mano de obra barata, escasamente calificada, exportador de materias primas o de bienes con escaso o nulo valor agregado debe ser sustituido por uno que permita competir en un mercado mundial globalizado con bienes y servicios de calidad, generados por recursos humanos competentes y bien remunerados que fortalezcan el mercado interno, al ofrecer alternativas laborales en todo el territorio para ampliar los niveles de bienestar de toda la población, lo cual cimentará un México con miras a integrarse a una economía globalizada y más justa para todos.

El desarrollo de la industria de software se presenta como una alternativa real para transformar aquellos estereotipos que han llevado a los clásicos fracasos comerciales:

- a) Ancla los recursos humanos en sus lugares de origen, así se reduce la fuga de cerebros que amenaza con esterilizar nuestras ciudades pequeñas y medianas.
- b) Genera empleos bien remunerados que fortalecen el mercado interno, un gran atractivo para la inversión.
- c) Produce y puede exportar productos y servicios de alto valor agregado.
- d) Genera divisas.

Sin embargo, el desarrollo de este segmento de la economía, actualmente en fase de crecimiento, se enfrenta a múltiples inhibidores, que están quemando el precioso y escaso tiempo del que disponemos para crear y fortalecer esta opción.

La naciente industria del software en México esta congelada y muy dispersa, donde prevalecen las estructuras administrativas familiares o informales, compuestas por un reducido número de profesionales, que en su mayoría y salvo excepción, están fuera de cualquier modelo de calidad reconocido internacionalmente (como ejemplo el Modelo de Madurez de Capacidades de la Universidad Carnegie Mellon, el modelo ISO 15504, entre otros).

La industria del software forma parte del grupo de actividades económicas que componen a las tecnologías de la información. Éstas se integran además a la industria del hardware y los servicios. Junto con las comunicaciones componen lo que se conoce como TIC (tecnologías de información y comunicación). La industria del software puede dividirse, en sentido amplio, en dos segmentos: el de aplicaciones en paquete y el de desarrollo de aplicaciones a medida.

México cuenta con una posición favorable para convertirse en un competidor de talla mundial en este ramo, gracias a su ubicación geográfica, perfil demográfico y estado de desarrollo tecnológico. No obstante el potencial de desarrollo es evidente, la industria de software es apenas incipiente en nuestro país: participa con tan sólo el 0.10% del PIB (cifras del 2000)<sup>5</sup>. Aunque no existe un patrón exhaustivo de esta industria que proporcione información exacta, una muestra de 206 empresas desarrolladoras de software muestra el perfil actual de la industria que es mayoritariamente micro y pequeña, con un tamaño muy inferior al del promedio internacional, que es de 250 empleados<sup>6</sup>, como simple referencia veamos la tabla 1.1 mostrada a continuación:

TAMAÑO	NÚMERO DE EMPLEADOS	PROMEDIO DE EMPLEADOS	NÚMERO DE EMPRESAS
Micro	Menos de 15	7	63
Pequeña	De 16 a 100	60	117
Mediana	De 101 a 250	175	14
Grande	De 251 a 1000	600	11
Corporativa	Más de 1000	1500	1

Tabla 1.1 Fuente: AMITI (Asociación Mexicana de la Industria de Tecnologías de Información, AC.)

Como usuario de TIC, México se sitúa en el lugar 50 a nivel mundial. Su gasto en este rubro en el 2001 equivalió al 3.2% del PIB nacional, proporción que no llega a la mitad de la que registro nuestro principal socio comercial, Estados Unidos con el 7.9%, y al promedio mundial, de 7.6%. En el caso de software, la diferencia es aún más dramática: 0.94% en Estados Unidos, 0.61% en el mundo y 0.10% en México. Esta cifra no guarda proporción con la posición de México en la economía mundial, en la que ocupa el décimo lugar.

<sup>5</sup>Fuente: Digital Planet: The Global Information Economy, Noviembre de 2000.

<sup>6</sup>Fuente: BANCOMEXT y AMITI.

## **1.5.2 Problemática del software en el México actual**

México se ha distinguido por los “bandazos” en su estrategia de desarrollo; de un modelo a otro término siendo que “la mejor estrategia era el que no existía”. Hoy en día, hay una clara visión de la importancia del software en el gobierno mexicano (Prosoft)<sup>7</sup> pero sigue haciendo falta una ejecución ágil de las políticas internas y mejores instrumentos para canalizar estímulos al sector productivo; corriendo el riesgo de quedar solo en buenas intenciones.

La industria mexicana de software es aún incipiente y ha evolucionado de manera lenta y con tropiezos. El mercado de software como producto está dominado por software internacional, aunque las aplicaciones para la gestión de la MPYMEs<sup>8</sup> son abrumadoramente mexicanas. No se realizan exportaciones importantes de software como producto.

Por otro lado, el mercado de software como servicio, si cuenta con exportadores, tanto de origen nacional como extranjero, aunque su volumen de negocios es aún bajo; las ventas totales de servicios de software no rebasan los 250 millones de dólares de facturación, con un contenido de 36 por ciento de exportación.

Una señal preocupante y que refleja la falta de madurez de nuestra industria de servicios de software es que más del 50 por ciento de la facturación es de suministro de personal (outsourcing), en donde el cliente controla los proyectos. Esto es el resultado de una mezcla compleja de factores; por un lado, la industria ha mostrado problemas en su capacidad de entrega de resultados en tiempo, calidad y costos y por otro, los clientes han puesto una atención excesiva en la reducción de los precios, forzando a los proveedores a reducir sus riesgos.

En conclusión, la industria actual no parece ser la base para iniciar un esfuerzo de exportación de alta envergadura. En nuestra opinión, sería más viable encontrar la fórmula para empatar la oferta y la demanda interna, promoviendo la profesionalización de ambas.

La capacidad de desarrollo de software cautiva en las organizaciones usuarias, dentro de los departamentos de sistemas, es muy superior a la misma industria y puede ser una plataforma más sólida para avanzar.

---

<sup>7</sup> Prosoft: Programa para el Desarrollo en la Industria de Software (<http://www.economia-sniim.gob.mx>).

<sup>8</sup> MPYMEs: Abreviación de micro, pequeñas y medianas empresas.

Los beneficios que se podrían obtener de invertir mejor estos recursos para incrementar la competitividad son superiores a las posibilidades inmediatas de exportación. Un aumento de la productividad de 2% con un buen aprovechamiento de la tecnología, equivaldría a más de 12,000 millones de dólares, más de 70 veces las exportaciones actuales de la industria.

México tiene la oportunidad de promover la industria del software adoptando un modelo propio, adecuado a sus circunstancias y sobre todo, a las perspectivas a futuro sobre el mercado; debe por lo tanto, abandonar la tentación de imitar a países como la India, Irlanda e Israel que han tenido éxito siguiendo modelos diseñados hace más de dos décadas. El foco de cualquier estrategia, tanto pública como privada, debe ser el desarrollo de una capacidad de integración y aprovechamiento de soluciones tecnológicas que contribuyan a la competitividad de la economía nacional.

La industrialización del software es necesaria e importante sólo en la medida en que facilita la capacidad de diseño e implantación de soluciones integrales de negocio; esta capacidad es escasa en el mundo, mientras que hay una oferta creciente de fábricas de software y una tendencia pronunciada hacia la baja de precios.

Existen varios puntos que por experiencia y conocimiento propios se han detectado a través del tiempo, considerando importante mencionarlos ya que se presentan como situaciones o problemas frecuentes en el desarrollo de sistemas en México:

- Hay un aumento constante del tamaño y complejidad de los programas.
- Dificultad de conseguir productos totalmente depurados, ya que en ningún caso un programa será perfecto.
- Se dedican elevados recursos monetarios a su mantenimiento, debido a la dificultad que los proyectos de software entrañan y a la no normalización a la hora de realizar los proyectos.
- No suelen estar terminados en los plazos previstos, ni con los costos estipulados, ni cumpliendo los niveles deseables de los requisitos especificados por el usuario.

- Incrementos constantes de los costos de desarrollo debido entre otros, a unos niveles de productividad bajos.
- Los clientes tienen una alta dependencia de sus proveedores por ser en muchos casos aplicaciones a "medida".
- Insuficientes procedimientos normalizados para estipular y evaluar la productividad, costos y calidad.
- Ausencia de especificaciones completas, coherentes y precisas previas por parte del cliente, así como posteriores por parte de los proveedores del software.
- Ausencia de la aplicación sistemática de métodos, procedimientos y normas de ingeniería del software.
- Escasez o ausencia de entornos integrados de programación.
- Escasez de uso de técnicas actuales y automatizadas para la gestión de proyectos.
- Escasez de personal con formación y experiencia en los nuevos métodos, normas y uso de entornos y utilidades de programación.
- Otros derivados del grado de desarrollo técnico y organizativo de cada compañía.

Como se puede observar existe una gran variedad de situaciones adversas que ponen en entre dicho la calidad quizá no sólo en el desarrollo de software, también en otras ramas que se ven afectadas por esta clase de circunstancias y para lo cual se puede notar que hay una labor titánica por realizar si es que México desea algún día pertenecer al "selecto" grupo de países desarrollados en tecnologías de información.

Quizá lo más difícil de realizar sea un cambio dentro de nuestra cultura tan llena de tradiciones que no son en su mayoría incorrectas, simplemente definen hábitos que sería mejor tratar de modificar y adaptar a la época actual para alcanzar los más altos niveles de desarrollo que se requieren hoy día ya que de lo contrario México tendrá que conformarse con hacer lo que dicten otras instancias alrededor del mundo y ser esclavos de la desidia.

## **1.6 El Modelo de Calidad como Estrategia de Solución**

En un concepto simple, un modelo es algo que por definición se debe seguir o imitar, a su vez un modelo no se puede entender si no se habla de una metodología que no es más que el modo de obrar o proceder de manera ordenada. Básicamente, un método que diga qué hacer en cada momento y con qué contenido; este conjunto de tareas organizadas y concretadas se conoce como metodología de desarrollo, y precisamente un modelo de calidad indica implícitamente que debe haber un método que sustente esa calidad y que sea en verdad tan bueno como para imitarse.

Una de las acepciones más utilizadas de la calidad es la relacionada con modelos de aseguramiento tipo CMMI (Capability Maturity Model Integration por sus siglas en inglés, en español se conoce como Modelo de Madurez de Capacidades Integrado) o ISO 9000. Haciendo uso de un modelo de este tipo se puede predecir la calidad, además tener control absoluto de las tareas para su realización, también saber en que punto de un proceso se produjo un error y subsanarlo, entre otras.

Incluso cada país suele tener su versión de metodología obligatoria (normalmente, en lo relativo a los aspectos formales orientados a la documentación) en los productos para la administración pública. La orientación adecuada consiste en partir de una metodología de desarrollo suficientemente contrastada y admitida, personalizada para la propia organización pero sin pérdida de la generalidad de la misma.

La elección de las herramientas adecuadas para el desarrollo de proyectos de calidad en una organización es fundamental para el éxito de dichas iniciativas. Generalmente, las organizaciones de clase mundial acuden a una combinación de modelos y metodologías para lograr resultados tangibles en su sistema de calidad, debido a que cada esquema provee elementos y enfoques diferentes que son complementarios entre sí.

En general los modelos de calidad definen a ésta de forma jerárquica, en otras palabras, la calidad se produce como consecuencia de la evaluación de un conjunto de indicadores o métricas en diferentes etapas.

En el nivel más alto de la jerarquía se encuentran los factores de calidad definidos a partir de la visión del usuario del software, y conocidos también como atributos de calidad externos.

Cada uno de los factores se descompone en un conjunto de criterios de calidad, o sea aquéllos atributos que cuando están presentes contribuyen a obtener un software de calidad.

Se trata de una visión de la calidad técnica, desde el punto de vista del producto software y se les denomina también atributos de calidad internos.

Finalmente para cada uno de los criterios de calidad se definen un conjunto de métricas o medidas cuantitativas de ciertas características que indican el grado en que dicho producto posee un determinado atributo de calidad.

De esta manera, a través de un modelo de calidad se concretan los aspectos relacionados con ella de tal manera que se puede definir, medir y planificar. Además el empleo de un modelo de calidad permite comprender las relaciones que existen entre diferentes características de un producto de software muchas veces complejo en primera instancia.

La cultura de calidad también se refleja en el incremento de las utilidades de una empresa. *“Existe un estudio que relaciona las utilidades de las organizaciones con el modelo de administración que emplean. Y los modelos de calidad total producen hasta un 350% más de utilidad, en relación a las que se administran con esquemas de supervisión tradicional”<sup>9</sup>.*

En el siguiente capítulo se hablará de un modelo que permite alcanzar de manera satisfactoria el aseguramiento de la calidad en el desarrollo de sistemas informáticos, por tratarse de un modelo extranjero se hace hincapié en el hecho de que se abordara de manera tal que, sea perfectamente comprensible no sólo para la gente de sistemas, sino para cualquier área que se interese en desarrollar productos de calidad y su repercusión invaluable en el desarrollo de un país; estamos seguros que el conocimiento así como la experiencia adquirida en este modelo servirán a estas y a futuras generaciones en la creación de un nuevo concepto y buen uso de metodologías de calidad aplicadas para el desarrollo de nuevos sistemas de información en México.

---

<sup>9</sup> José Ramón Sida, director general de la “Fundación Mexicana para la Calidad Total”, charló con usuarios de [www.intermanagers.com](http://www.intermanagers.com) el 31 de mayo de 2004 sobre “Cómo dirigir una organización con el modelo de calidad”.

# Capítulo 2



En el capítulo anterior se habló de la calidad y la importancia que tiene en toda organización o empresa que desea competir dentro del mercado mundial, el cual al paso del tiempo se ha convertido en un proceso mucho más complicado; hoy día la calidad esta inmersa en la mayor parte de los productos o servicios y no es posible separar una cosa de la otra, por esto mismo resulta necesario tratar el tema detalladamente y comprender los alcances que la calidad y una buena estrategia aplicada pueden lograr en cualquier organización que es lo que dio origen al modelo CMMI (Capability Maturity Model Integration por sus siglas en inglés, en español se conoce como Modelo de Madurez de Capacidades Integrado), se hablará de su conformación y las bases en las que se sustenta su éxito. La mayor parte de las empresas están interesadas en mantener siempre el control de lo que hacen en todas sus áreas productivas y con esto garantizar la calidad de sus productos sin importar el tipo o la complejidad de los que éstos sean, precisamente es aquí donde se encuentra una oportunidad muy grande de manejar un modelo que permita tener dicho control y que a manera de prácticas fundamentales al ser incluidas dentro de la organización, coadyuven a lograr los objetivos planteados en los tiempos y costos mínimos.

## Objetivo

Dar a conocer los fundamentos principales del modelo CMMI y su uso enfocado al desarrollo de software para asegurar la calidad total, así mismo aprender a usar mejor los recursos durante un proyecto de desarrollo. Hacer conciencia en las personas y organizaciones que desarrollan sistemas en tomar una iniciativa para comenzar un mejoramiento de la madurez de sus procesos, si es que desean aumentar su productividad y la calidad de sus productos.

## **Justificación**

Los modelos de calidad existen para ayudar a desarrollar y crear productos de primer nivel con un esfuerzo mejor distribuido, optimizando los recursos y cumpliendo los objetivos en el menor tiempo posible, el presente trabajo habla del modelo CMMI y propone una manera correcta de aplicarlo en todas aquellas empresas que deseen llegar a ser más productivas con miras a competir en el mercado mundial y que basan su desarrollo en sus sistemas y en la manera en las que estos operan.

En nuestro país el desarrollo de software tiene un enfoque distinto al de otros países, aquí propiamente se usan y mantienen los sistemas importados de naciones como Estados Unidos, España y algunos otros en menor escala y pocas veces una empresa mexicana se preocupa por desarrollar algo desde cero. Pero esto tiene una explicación básica, México tiene un problema severo desde hace muchos años y que tiene que ver con la cultura del esfuerzo y el trabajo.

La manera en la que los ingenieros de sistemas trabajan actualmente, dista mucho de la labor que se debería estar realizando, ya que en la mayor parte de las empresas el desarrollo es muy mínimo y no figura como un aspecto comercial importante, si a esto se suma que ciertas empresas nacionales tienen convenios millonarios con empresas extranjeras de software, resulta más complicado mirar hacia adentro y darse cuenta, que mientras más se deja en manos de otros lo que como ingenieros en computación o sistemas se debería estar realizando, llegando a un punto crítico de dependencia absoluta que sólo llevará lamentablemente a una crisis de software aún peor de la que ya existe.

Con el fin de hacer conciencia de esto, en este capítulo se abordará el modelo CMMI, que a pesar de haber sido realizado en su totalidad por una universidad extranjera, se considera que esta muy bien sustentado en los principios fundamentales de la ingeniería y su filosofía no esta encasillada en el modelo americano, por el contrario contempla todas y cada una de las facetas propias del desarrollo y control de procesos de software más comunes en el mundo.

Se considera que el uso de este modelo en cualquier empresa puede garantizar la calidad en el desarrollo de proyectos y sistemas; y por supuesto se hablará de su relación y relevancia en el desarrollo de software dentro de una institución financiera.

## 2.1 Modelos y estándares de calidad

La figura 2.1 es un mapa de situación simplificado con la foto de los principales marcos y modelos de procesos que pueden servir de referencia a las organizaciones de software; y las principales referencias de sus orígenes y evolución.

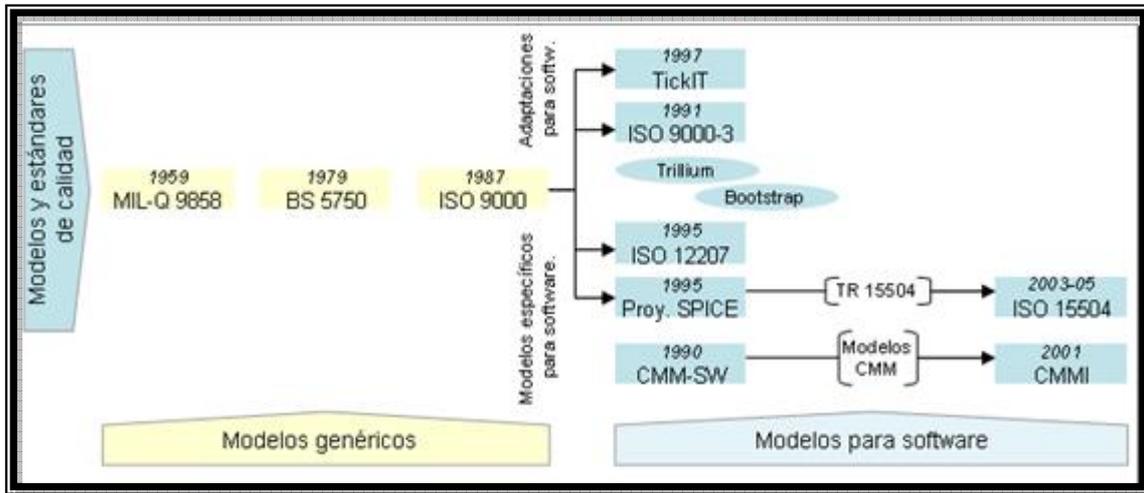


Figura 2.1 Modelos y estándares de calidad

Los procesos de fabricación de los años 50 hicieron necesaria la normalización de los procesos de producción para garantizar la consistencia de los resultados sobre unos parámetros o requisitos previamente determinados.

La norma que se considera como punto de arranque de los posteriores estándares, marcos y modelos que se han extendido a todas las industrias es la desarrollada por EE.UU. en 1959: “Quality Program Requirements” MIL-Q-9858, que, inicialmente diseñada para el ámbito militar, estableció un esquema auditable (a través de la norma de inspección MIL-I.45208) de los requisitos que los proveedores debían cumplir.

El uso de esta norma se fue generalizando, pero de forma paralela diferentes países y organizaciones comenzaron a desarrollar las suyas propias. Así por ejemplo la OTAN en 1968 adoptó las especificaciones AQAP “Allied Quality Assurance Procedures”, en español se conoce como “Procedimientos aliados en el aseguramiento de la calidad”.

El estándar británico BS5750, adoptado en el Reino Unido en 1979 fue el siguiente hito relevante en el camino de las normalizaciones, al lograr gran reconocimiento en varios países. Esta normativa fue en realidad la precursora de ISO 9000.

Las normas y estándares que fueron surgiendo de los años 60 a los 90 dieron respuesta a las garantías de homogeneidad, calidad y de cultura de prevención en los entornos de fabricación.

Algunos de los fracasos que se produjeron en determinados sistemas por la introducción del software sin que éste pudiera dar garantías de resultados a la altura de ingenierías más maduras, son ya parte de la historia de la profesión.

Esbozar este marco histórico resulta necesario, porque es la causa de la actual sopa de letras capaz de marear al gestor más dispuesto con sólo enumerarla: ISO 9000-3, TicIT, ISO 12207, CMM-SW, ISO 15504, CMMI, ASD, XP, SCRUM, DSDM, PP, MSF, etc.

Tomando un poco de perspectiva (tan sólo 3 décadas), al software se le ha juntado la necesidad de encajar en marcos de calidad para ofrecer garantías de perfección, eficiencia y consistencia en los resultados; con la creación de su propia base de conocimientos técnicos para desarrollar y mantener software (requisitos, codificación, pruebas, diseño, etc.)

En este punto es interesante diferenciar entre marco o modelo de procesos y técnica o base de conocimiento técnico. La primera línea tiene como objetivo fijar qué es lo que hay que hacer para ofrecer garantías en los proyectos, y la segunda cómo debe realizarse.

Un marco de procesos determinará, por ejemplo, la necesidad de gestionar adecuadamente las modificaciones de requisitos, o de realizar un diseño detallado antes de comenzar la codificación. El valor de los modelos de procesos radica en el conocimiento que aportan al señalar las actividades cuya omisión incrementa las posibilidades de fracaso en los proyectos.

Sin embargo el marco de procesos no dirá, por ejemplo, que deba emplearse un sistema basado en documentos o en base de datos para la gestión de los requisitos; o si el diseño debe realizarse con diagramas IDEF o UML, esté es el campo de la técnica de la ingeniería del software.

En la línea de los modelos de procesos, a partir de finales de los 80 se comenzaron a desarrollar marcos específicos para el software, para dar la cobertura necesaria a las particularidades del producto, donde las normas generales se quedaban cortas.

En esta línea ISO 9000 desarrolló una norma específica para el software: ISO 9000-3, y la BSI (British Standards Institution) hizo lo propio desarrollando TickIT.

En esta primera aparición de estándares surgieron también, aunque con menor repercusión: Trillium y Bootstrap. El primero desarrollado por la empresa Bell Canadá, que liberó sus derechos, haciéndolo de dominio público. Bootstrap es una metodología para la evaluación, medición y mejora de los procesos de software. Su desarrollo lo llevó a cabo una comisión del ESPRIT (European Strategic Program for Research).

Sin embargo las dos líneas que surgieron a principios de los 90 y que continúan como referentes en la actualidad son CMM (actualmente conocido como CMMI), y las normalizaciones de ISO 12207 y 15504.

## **2.2 Metodología de desarrollo de software**

Las organizaciones que desarrollan o mantienen software pueden optar por trabajar "a la buena de Dios", o por seguir una metodología. Hacerlo a la buena de Dios no es tan raro. Es la primera forma que se empleó para desarrollar programas.

Las metodologías de desarrollo son un conjunto de procedimientos, técnicas y ayudas a la documentación para el perfeccionamiento de productos especializados de software.

Es como un libro de recetas de cocina, en el que se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Actualmente es imprescindible considerar los riesgos, aunque habitualmente las empresas, no han sido sensibilizadas de los riesgos inherentes al procesamiento de la información mediante computadoras, a lo que han contribuido, a veces, los propios responsables de informática, que no han sabido explicar con la suficiente claridad las consecuencias de una política de seguridad insuficiente o incluso inexistente.

Por otro lado, debido a una cierta deformación profesional en la aplicación de los criterios de costo/beneficio, el directivo desconocedor de la informática no acostumbra a autorizar inversiones que no lleven implícito un beneficio demostrable, tangible y mensurable.

Las técnicas indican cómo debe ser realizada una actividad informática determinada identificada en la metodología. Combina el empleo de unos modelos o representaciones gráficas junto con el empleo de unos procedimientos detallados. Se debe tener en consideración que una técnica determinada puede ser utilizada en una o más actividades de la metodología de desarrollo de software.

### 2.2.1 Importancia en el uso de una metodología específica

Hay un gran número de factores que repercuten en la persona que trabaja dentro de un entorno de desarrollo software. Los cambios en el sistema operativo, el lenguaje de programación, la organización del proyecto, o los estándares establecidos para los diferentes aspectos del ciclo de vida de un proyecto pueden influir tanto en el trabajador como en la cantidad de trabajo que puede realizar.

La productividad, como una medida cuantitativa de la cantidad de trabajo que puede ser realizada por una persona, se puede alterar de distintas maneras, algunas de ellas tan simples como, por ejemplo, enseñar a todos los implicados en el trabajo a escribir a máquina. Este hecho, sin ir más lejos, podría tener un mayor impacto en la productividad que el de introducir nuevas herramientas de software o técnicas de diseño.

Sin embargo la productividad no tiene en consideración la calidad del producto. Por ejemplo los trabajadores en una planta de ensamblaje de computadoras pueden producir 100 de éstas por hora, pero la medida no es útil, en cuanto que los equipos pueden requerir trabajo adicional para corregir problemas surgidos en la etapa del ensamblaje. Lo mismo ocurre en el desarrollo de software, el objetivo es establecer un entorno que no sólo mejore la productividad del que lo desarrolla, sino que también genere la creación de mejores productos.

Es obvio que el elemento más importante en cualquier empresa de desarrollo de software es disponer de personas con una elevada preparación, y sin embargo ello no asegura el éxito en la consecución de los objetivos propuestos, ya que existe el peligro de una falta de conjunción, producida por la manera personal de desarrollar el software de cada individuo, por muy bueno que éste sea, y la imposibilidad de un auténtico trabajo en equipo.

El Modelo de Madurez, indica que los mejores informáticos necesitan un entorno disciplinado y estructurado en el cual puedan realizar un trabajo en equipo, para lograr productos con alta calidad.

El ingeniero de software es una persona que trabaja en equipo, que conoce que lo que realiza es un componente que se combinará con otros para formar un sistema. Es consciente de que el componente software que diseña debe poseer los principios de la Ingeniería del Software para que el sistema final sea satisfactorio.

Los programadores tradicionales argumentan que la aplicación de una metodología supone una gran carga. Es cierto, pero si no se emplea una metodología pueden surgir los siguientes problemas:

- La introducción de nuevas herramientas que afectarán al proceso.
- Resultados distintos a los esperados.
- Detección tardía de errores.
- Resultados impredecibles.
- Cambios de organización.

La situación actual se debe ver como una situación en la que la empresa que comience a poner los elementos necesarios para mejorar el proceso software tendrá mucha más ventaja competitiva frente a las demás.

Sólo como referencia, en países como la India, la industria del software es una de las más desarrolladas del mundo y la mayoría de sus organizaciones de software está en los niveles más elevados de calidad. Sus exportaciones superan los cinco mil millones de dólares anuales y el 65% de producción lo compra Estados Unidos. La clave está en la calidad y en la madurez de sus organizaciones y procesos.

## 2.3 Mejoramiento de procesos

El desarrollo de productos de software tiene tres grandes componentes:

- **Personal.**

Incluye el conocimiento y experiencia del capital humano que crea y sostiene la evolución del producto. Sin el personal competente y experimentado, es imposible crear productos que satisfagan las necesidades de los clientes.

- **Tecnología.**

Incluye la posesión de las tecnologías que sustentan el producto y las herramientas utilizadas en su desarrollo.

- **Proceso.**

Es el saber como utilizar el conocimiento del personal y la tecnología en forma eficiente para lograr productos de alta calidad que satisfagan las necesidades de los clientes, producidos dentro de costos y plazos aceptables.

La siempre creciente demanda de las empresas desarrolladoras de software ha producido una crisis en la disponibilidad de ingenieros dedicados al ramo en el mercado laboral. Los recursos humanos, cada vez más caros y escasos, deben ser utilizados de manera eficaz y productiva. Si bien es cierto que el costo de las herramientas para producir software (computadoras y software de desarrollo) han tenido una tendencia a la baja, la creciente complejidad de la tecnología que se debe incorporar en los productos se ha encarecido (e.g. aumento del costo de investigación y desarrollo, patentes, etc.)

El proceso representa una fuerte inversión en recursos y tiempo. La construcción de software implica una larga incubación estrechamente ligada a la cultura de la organización y un enorme esfuerzo de experimentación y errores.

Un proceso inadecuado puede tener graves consecuencias y acarrear costos intolerables, lo cual puede significar la diferencia entre el éxito y el fracaso en el competitivo mercado de nuestros días.

### 2.3.1 ¿Qué significa mejorar el proceso?

En vez de proponer una reingeniería radical de los procesos y competencias existentes en la empresa, habitualmente de enorme costo y alto riesgo, se parte de la base que existe un interés genuino de los ingenieros y gerentes por crear procesos maduros, que permitan usar adecuadamente sus talentos y los recursos asignados. Ambos buscan minimizar los problemas evitables y fortalecer la prosperidad común que resulta del éxito de la empresa.

A través de los años los empresarios han manejado sus negocios trazándose sólo metas limitadas, que les han impedido ver más allá de sus necesidades inmediatas, es decir, planean únicamente a corto plazo; lo que conlleva a no alcanzar niveles óptimos de calidad y por lo tanto a obtener una baja rentabilidad en sus negocios.

Según los grupos gerenciales de las empresas japonesas, el secreto de las compañías de mayor éxito en el mundo radica en poseer estándares de calidad elevados, tanto para sus productos como para sus empleados; por lo tanto el control total de la calidad es una filosofía que debe ser aplicada a todos los niveles jerárquicos en una organización, y está implica un proceso de *mejoramiento continuo* que no tiene final. Dicho proceso permite visualizar un horizonte más amplio, donde se buscará siempre la excelencia y la innovación que llevará a los empresarios a aumentar su competitividad, disminuir los costos, orientando los esfuerzos a satisfacer las necesidades y expectativas de los clientes.

Para llevar a cabo este proceso de mejoramiento continuo tanto en un departamento determinado como en toda la empresa, se debe tomar en consideración que dicho proceso debe ser: económico, es decir, debe requerir menos esfuerzo que el beneficio que aporta; y acumulativo, que la mejora que se haga permita abrir las posibilidades de sucesivas mejoras a la vez que se garantice el cabal aprovechamiento del nuevo nivel de desempeño logrado.

El mejoramiento de procesos de software usa metodologías prácticas basadas en la experiencia colectiva de la industria de software internacional.

Uno de los estándares de facto más importantes es el modelo de madurez de capacidades (CMMI), que ayuda notablemente a mejorar la calidad de los procedimientos dentro de una organización y que impacta en los resultados económicos que son el fin de toda empresa.

## 2.4 Modelo CMMI (Capability Maturity Model Integration)

Se inicia por definir el significado de **CMMI**, es una abreviación en idioma inglés que significa **M**odelo de **M**adurez de **C**apacidades **I**ntegrado, sus orígenes datan del año 1991, desarrollado por el "Software Engineering Institute" (SEI) de la Universidad Carnegie Mellon en Pittsburg.<sup>10</sup>

Es un modelo de prácticas fundamentales que toda organización interesada en desarrollar y mejorar la calidad de sus productos debería implementar, este modelo está basado en conceptos de calidad total así como de mejoramiento continuo y ha sido ampliamente aceptado por la comunidad de ingeniería de software. Rápidamente se ha convertido en el estándar de facto en los Estados Unidos de América, y también a nivel internacional, para evaluar la madurez de procesos que tienen las organizaciones que producen software.

Este modelo se puede utilizar no solamente como un manual de prácticas recomendables, sino como referencia para llevar a cabo auditorias y evaluaciones internas en las organizaciones.

El SEI ha desarrollado algunos métodos de evaluación basados en este modelo. Uno de estos métodos es la evaluación basada en CMMI para el mejoramiento interno de procesos, generalmente conocido como CBA IPI "CMM - Based Assessment Internal Process Improvement" (CMM – Evaluación basada en el mejoramiento de los procesos internos); aunque su principal objetivo es permitir a la empresa la determinación de sus fortalezas y necesidades de mejoramiento, también permite revisar las prácticas de los proveedores externos, a objeto de que puedan derivar un plan de mejoramiento adecuado a su organización.

Personas tales como Deming, Juran, Crosby y otros han advertido sobre la necesidad que tienen las compañías de aplicar un enfoque de calidad total. Cualquiera que nunca haya escuchado la expresión "Calidad Total" probablemente desconoce la relevancia que tiene esta expresión en cualquier empresa. Toda organización respetable tiene su plan de calidad y cuando las iniciativas comienzan en el departamento de sistemas, tanto gerentes como profesionales se preguntan a menudo como aplicar dicha filosofía en su área de especialización.

---

<sup>10</sup> Fuente: página oficial del SEI (<http://www.sei.cmu.edu/cmmi/models/models.html>).

### 2.4.1 Breve Historia de Desarrollo

Desde 1991, se han desarrollado CMM's en una gran variedad de disciplinas. Algunas de las más notables incluyen modelos para la ingeniería de sistemas, ingeniería de software, adquisición del software, dirección de la mano de obra y desarrollo, e Integración de Productos y Desarrollo de Procesos.

Aunque estos modelos han demostrado una gran utilidad en muchas empresas, el uso de algunos de ellos ha sido problemático. A una gran cantidad de organizaciones les gustaría enfocar sus esfuerzos de mejoramiento a través de ciertas disciplinas. Sin embargo, las diferencias entre estos modelos o disciplinas específicos, incluyendo su arquitectura, volumen, y proximidad han limitado la habilidad de enfocar sus mejoras con éxito. Más allá de esto, la aplicación de múltiples modelos que no llegan a integrarse correctamente en una organización conlleva más costos relacionados a entrenamiento, productividad, y actividades de mejoramiento. Una selección bien integrada de modelos y enfocada eficazmente a múltiples disciplinas así como un entrenamiento bien organizado resuelve estos problemas.

El proyecto CMMI comenzó formalmente su desarrollo en el año 1997 patrocinado por la OSD (Office of the Secretary of Defense) y la NDIA (National Defense Industrial Association) en asociación con el SEI, esto con el objetivo de conformar un solo modelo integrado para solucionar el problema de usar múltiples modelos y las confusiones originadas por su similitud. Para cumplir con esto, se combinaron tres ya consolidados que sirvieron de base, el Modelo de Madurez de Capacidades enfocado al Software (SW-CMM v2.0), la Alianza de Industrias Electrónicas Provisionales Estándar (EIA/IS 731), y el modelo de Desarrollo de Productos Integrados del CMM (IPD-CMM v0.98), todos incluidos en un solo esquema de mejoramiento para el uso de organizaciones grandes y con la firme convicción de alcanzar niveles de calidad internacionales.

La integración de varios modelos no es sencilla, el gran reto para el equipo de trabajo que se encargó de elaborar dicha estrategia fue el hacer que todos los productos desarrollados fueran consistentes y compatibles con la Organización Internacional para la Estandarización / Comisión Internacional Electrotécnica (ISO/IEC).

El CMMI desarrollado por el SEI en colaboración con la comunidad de sistemas satisface la necesidad de aplicar los principios de calidad total específicamente al desarrollo de software.

Por consiguiente, este modelo sirve como una guía para el mejoramiento del desarrollo y mantenimiento de las prácticas de software. Consiste en prácticas clave que expresan métodos óptimos para realizar productos de calidad, con aumento de productividad, con un presupuesto y fechas previstas. Estas prácticas se identificaron al cabo de una amplia consulta con la comunidad de software, basándose en los éxitos y fracasos experimentados en miles de proyectos pasados y actuales.

El SEI se niega a hacer del CMMI un estándar, insiste en la necesidad de que cada organización interprete las prácticas que la conforman con respecto a su propia situación. No es una Biblia o una fórmula mágica. No se pretende que sea un remedio universal para todos los males de las empresas que trabajan con software. Es, reiteramos, un modelo, y las ventajas y limitaciones de dicho modelo se explicarán en este capítulo.

## **2.5 Seleccionando un modelo CMMI**

Existen múltiples modelos CMMI disponibles, por consiguiente, es muy importante analizar y conocer más a detalle qué modelo se ajusta mejor a las necesidades de la empresa u organización para optimizar y mejorar sus procesos.

La diferencia radica en identificar cual es la verdadera necesidad del negocio, aunque existan cuatro modelos y que sean tan similares no significa que no se haga necesaria la correcta explicación de lo que cada uno es capaz de abarcar.

Esto se hace con la finalidad de escoger un modelo que cubra aspectos que a la empresa le sean relevantes, se debe elegir el más óptimo y apegado a los requerimientos ya que de ello dependerá el éxito o el fracaso del proyecto donde se este implantando.

Actualmente existen cuatro modelos del CMMI como ya se habían mencionado, también conocidos como “disciplinas” pero ahora se explicarán con más detenimiento para comprender sus diferencias y características particulares, aunque por el momento sean sólo cuatro el equipo de productos CMMI prevé integrar algunos nuevos en un futuro.

Para un mejor enfoque y correcta aplicación, el CMMI se encuentra “dividido” actualmente en cuatro disciplinas que la integran y que en realidad comparten cierta similitud, pero se han fraccionado para facilitar su aplicación.

### **2.5.1 Systems Engineering (Ingeniería de Sistemas)**

La disciplina de Ingeniería de Sistemas como su nombre lo indica cubre el desarrollo total de los sistemas, el cual podría o no incluir los aspectos relacionados con el software. Se enfoca sobre la transformación de las necesidades del cliente, sus expectativas y limitaciones dentro de las soluciones de producto y contempla el soporte necesario durante la vida de dicho producto. Cuando se selecciona el modelo de Ingeniería de Sistemas, se considera el control del proyecto, soporte e ingeniería en las áreas del proceso así como su gestión.

### **2.5.2 Software Engineering (Ingeniería de Software)**

La ingeniería de software cubre el desarrollo de sistemas basados precisamente en el enfoque de software sobre la aplicación sistemática, disciplinada y propuestas cuantificables para el desarrollo, operación y mantenimiento del software. Cuando se selecciona este modelo, se considera la gestión del proceso, el control del proyecto, soporte e ingeniería en cada una de las áreas relacionadas al desarrollo.

### **2.5.3 Integrated Product and Process Development (Desarrollo integrado de productos y procesos)**

Este modelo es un acercamiento sistemático que logra una colaboración oportuna con todas las áreas de desarrollo a lo largo de la vida del producto para satisfacer las necesidades del cliente, sus expectativas, y requisitos. Se trata de integrar procesos nuevos enfocados para trabajar conjuntamente con los procesos que ya existen en la organización. Esta disciplina por si sola no genera todos los resultados esperados, si una empresa u organización desean desarrollar productos se necesita usar esta disciplina en colaboración como ya se menciona con alguna otra disciplina para obtener los resultados esperados.

## 2.5.4 Supplier Sourcing (Proveedor de Recursos)

Cuando los esfuerzos en el trabajo se tornan más complejos, lo común es atacar los proyectos haciendo uso de proveedores para realizar las funciones o agregar las modificaciones a los productos que se necesitan específicamente por el proyecto. Cuando esas actividades son críticas, el proyecto se beneficia del análisis y el monitoreo de los proveedores antes de entregar el producto final. Esta disciplina cubre la adquisición de productos de los proveedores bajo estas circunstancias de trabajo arduo y complejo.

Cuando se opta por esta disciplina, el modelo contendrá la dirección del proceso, dirección del proyecto, apoyo y áreas de proceso de ingeniería que aplican los recursos del proveedor. La dirección integrada del proveedor esta incluida bajo el área de dirección del proyecto y todo lo que maneja esta categoría dentro del modelo esta presente en otras áreas del proceso para ayudar a interpretar las prácticas específicas para el proveedor.

## 2.6 Usos sugeridos de este modelo

Puede usarse ciertamente como una referencia formal para las auditorias de proveedores, como un medio de evaluar su habilidad de llevar a cabo un determinado proyecto computacional. También puede ser usado por una organización para su propia evaluación interna de la madurez del proceso con el objeto de desarrollar un plan de mejoramiento.

Debido a que el CMMI es un modelo altamente detallado de buenas prácticas, también proporciona muchas ideas que pueden ser fácilmente puestas en marcha con el uso del mismo. Las prácticas descritas en él, van acompañadas de ejemplos y sugerencias, lo que también lo hace útil para aquellos que se encuentran implementando acciones de mejoramiento específicas.

Este modelo es muy completo e interesante ya que al estar dividido en diferentes niveles como se verá más adelante, permite atacar cada problema desde sus diferentes ángulos, y no deja nada al azar, se puede estar seguro que su implantación aunque requiere de personal capacitado es fácilmente adoptado por los miembros de un equipo de trabajo. Finalmente, de una manera más general, se recomienda como una herramienta de referencia para los profesionales y gerentes involucrados en calidad de software.

El CMMI no es aplicado generalmente en las grandes organizaciones, como en principio se podría creer, la tendencia está en buscar que un gran número de empresas pequeñas lo pueden implantar, aunque el gran obstáculo para la aplicación de este modelo son los costos.

Entre sus características se pueden mencionar:

- No es una estructura rígida, provee un esquema de trabajo consistente que permite nuevas iniciativas.
- Se enfoca en el problema del sistema Total a diferencia de SW-CMM O SE-CMM.
- Facilita la mejora en los procesos de toda la empresa.
- Permite contar con un proceso sistematizado.
- Mejora la productividad en el desarrollo y gestión de proyectos, con un mayor ajuste a los plazos y costos esperados.
- Entre CMM (antecesor del modelo actual) y CMMI una de las diferencias que existe es que en CMM existen cinco niveles para clasificar a las organizaciones respecto a la madurez de sus procesos, cada uno de los cuales es una plataforma y requisito para avanzar al siguiente; mientras que en la metodología CMMI, tiene dos formas de representación: por estados, en el cual hay niveles consecutivos y representación continua, que permite que aquellas empresas mejoren sólo en alguna área de sus procesos.
- Los procesos de evaluación de ambos son muy similares, aunque CMMI, tiene un mayor grado de exigencia, al igual que el trabajo de recolección de evidencias y preparación de documentación.

## 2.7 Componentes del modelo CMMI

El modelo CMMI está diseñado para medir la madurez de los procesos de software dentro de una organización así como para identificar las prácticas claves necesarias para incrementar la madurez de dichos procesos.

Dentro de este modelo se describen los principios y prácticas que definen la madurez del proceso de desarrollo de software y ayuda en gran medida a las empresas u organizaciones a gestionar en términos de evolución desde procesos caóticos a procesos disciplinados.

En la figura 2.2 se presenta de manera gráfica los componentes del modelo CMMI.

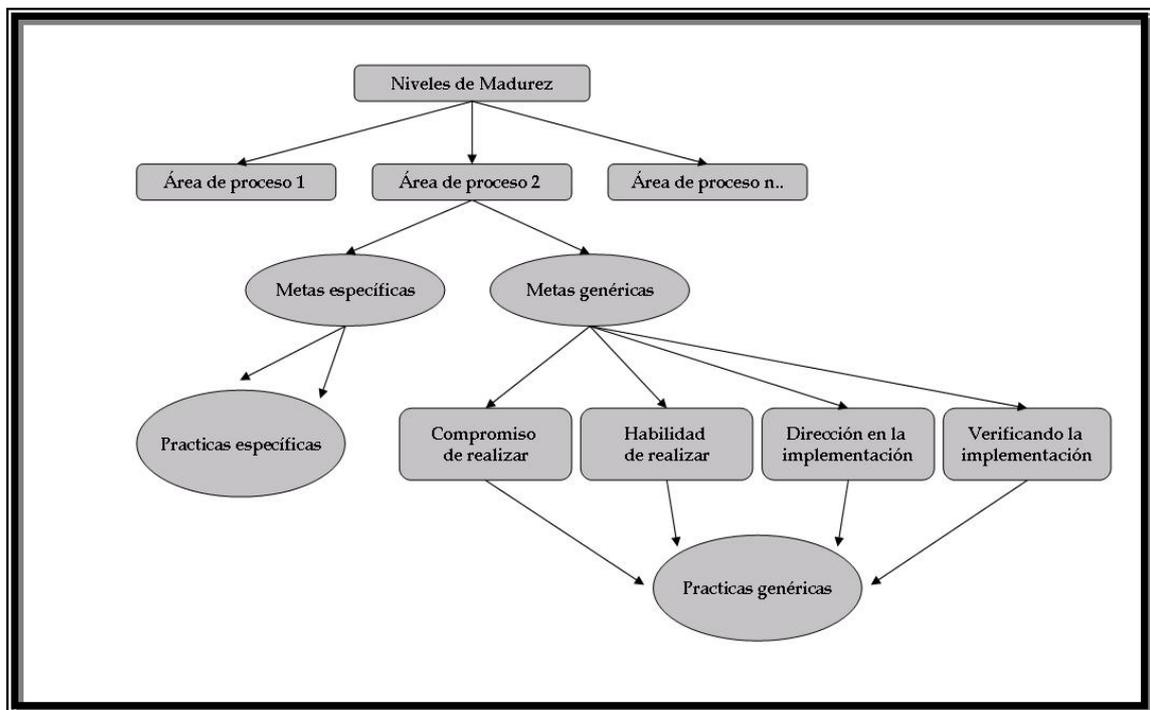


Figura 2.2 Componentes del modelo CMMI

Como puede observarse, este modelo se basa en niveles de madurez que son usados como un criterio para medir los métodos y mecanismos de una organización de software. Dentro de cada nivel de madurez se enfatizan distintos puntos que se redefinen como áreas del proceso (también llamadas áreas clave) que son distintas para cada nivel pero que deben ser completadas para poder pasar al siguiente nivel de madurez. Cada área de proceso tiene en si mismo metas específicas y genéricas, a su vez las metas específicas contienen prácticas específicas y las metas genéricas están organizadas en cuatro características comunes las cuales constituyen atributos que indican si la implantación y la institucionalización de un proceso clave es efectivo, repetible y duradero.

### **2.7.1 Niveles de madurez**

La creciente necesidad, sumada a décadas de promesas incumplidas en cuanto a calidad y costos en el desarrollo de software, condujo al Instituto de Ingeniería del Software (SEI) de la Universidad Carnegie Mellon en Pittsburg a desarrollar un método para evaluar el nivel de madurez del proceso de desarrollo del software de una empresa. El proceso se evalúa mediante un cuestionario y las respuestas sirven para determinar una magnitud denominada "Nivel de Madurez del Proceso".

En principio, el CMMI fue creado para evaluar y mejorar la capacidad de los contratistas de software del Departamento de Defensa de los Estados Unidos, este modelo se convirtió a través de los años en el más alto estándar de ingeniería en el mundo para todo tipo de compañías. Está fundamentado en prácticas reales de las compañías más avanzadas, y refleja lo mejor en procesos de desarrollo de software.

El CMMI está compuesto de 316 prácticas claves agrupadas en 18 áreas y distribuidas en una jerarquía de 5 niveles, a través de los cuales una organización progresivamente alcanza mayor calidad, productividad y menores costos en el desarrollo de software. Los niveles progresan desde el 1, que representa el estado caótico, hasta el nivel 5, que representa el estado de optimización continua.

La experiencia ha mostrado que las organizaciones logran mucho más cuando enfocan sus esfuerzos durante el proceso de mejora continua en un número manejable de áreas del proceso que requieren el esfuerzo en aumento conforme la organización va mejorando.

Un nivel de madurez es una meseta evolutiva definida de mejora del proceso. Cada nivel de madurez estabiliza una parte importante de los procesos de la organización.

El valor obtenido es un indicador de toda la empresa, aunque puede darse el caso de que en algún departamento se tenga un nivel de madurez mayor o inferior al resultante general. Los niveles de madurez del proceso son cinco como ya se mencionó y se definen a continuación.

### **2.7.1.1 Nivel 1 o Inicial**

El nivel 1 o inicial es el estado primario en la evolución de las organizaciones que desarrollan software.

Una definición amplia es que en este nivel se encuentran todas las empresas que no han logrado implementar las prácticas básicas de gestión de proyectos e ingeniería de software definidas a partir del nivel 2 o superiores. Una empresa está en el nivel caótico cuando sus gerentes y personal afirman que los proyectos no se pueden planear, que los requerimientos no se pueden tener bajo control, que no se esté siempre en condiciones de controlar las versiones del producto, donde la calidad sea percibida como una burocracia innecesaria, cuando se acepte que los procesos son una cosa personal, cuando no se pueda verificar ni validar el producto, y sobre todo, cuando sus gerentes y personal vivan bajo condiciones de stress y frustración permanentes.

En este tipo de empresas, el software es virtualmente producto del arte más que de la ingeniería. Cada "artista" crea su propio proceso, el cual es parte de su sello personal. La gerencia ocupa una parte significativa de su tiempo en atacar problemas y enfrentar clientes insatisfechos. Ante una situación de crisis permanente, se les hace difícil destinar recursos para definir o documentar procesos, lo que lleva a un círculo sin salida.

Cuando el proyecto se termina, la inversión hecha en desarrollar el proceso es raramente reutilizada en nuevos proyectos. Los desarrolladores de software generalmente tienen que trabajar largas horas y enfrentar problemas en forma cotidiana, lo cual les disminuye su creatividad y productividad netas. El éxito descansa en los hombros de estos héroes, tal como en una película de acción americana. Su nivel de frustración es elevado y es muy frecuente que decidan explorar caminos en otras empresas con menor nivel de stress. El proceso, que no está documentado ni ha sido compartido, se va con ellos. Los reemplazantes heredan problemas y dificultades, pero son raramente capaces de recuperar los procesos de desarrollo. Esto obliga a reinventar la rueda, a un alto costo y retrasando los proyectos. Algunas empresas han llegado a la conclusión que es demasiado caro o difícil tratar de adivinar lo que el empleado anterior hizo, les resulta mejor echar a la basura el desarrollo previo y empezar todo de nuevo. En casos extremos hay que simplemente terminar el producto para no seguir perdiendo dinero o prestigio frente a los clientes.

### 2.7.1.2 Nivel 2 o Repetible

El nivel 2 o Repetible hace posible la implantación de prácticas mínimas de administración de proyectos, de control de requerimientos, versiones de producto y de proyectos realizados por subcontratistas. El grupo o equipo humano que realizó el proyecto puede aprovechar su experiencia e inversión en procesos para aplicarla en un nuevo proyecto.

En esta etapa la organización ha logrado las metas genéricas y específicas, en otras palabras, los proyectos se han asegurado en cuanto a que los requerimientos están bien manejados y los procesos correctamente planificados, medidos y controlados.

Este nivel no garantiza que todos los proyectos dentro de la empresa estén necesariamente al mismo nivel de madurez. Algunos pueden estar todavía en el nivel inicial. Se han visto algunos casos en la práctica y en todos ellos estos proyectos fueron ineficientes con respecto a los de mayor madurez, malgastando el éxito de estos últimos. Eventualmente algunos invirtieron los esfuerzos necesarios para ponerse a tono, otros simplemente fueron cerrados con un elevado costo de frustración y descalabro de carreras profesionales.

#### **Beneficios de implantar el nivel 2.**

El mayor beneficio que se obtiene de la implantación del nivel 2 por las empresas, es la planificación realista de los proyectos. Antes los cronogramas<sup>11</sup> de cada proyecto expresaban más los deseos de la gerencia que la realidad. Este principio (el mismo en la cual se basa la magia) conducía una situación de buscar culpables y generar excusas, produciendo al mismo tiempo frustración y desconfianza entre clientes y empleados. Actualmente los cronogramas son cada día más confiables, y mejoran a medida que se acumula más información en las bases de datos de los proyectos pasados. El uso generalizado de métodos de estimación permite al personal del proyecto justificar plazos y recursos. Aún el "olfato profesional" y la experiencia personal juegan un papel importante en la generación de planes de proyecto, pero ahora son decisiones informadas en vez de simples adivinanzas como en el pasado.

---

<sup>11</sup> Cronograma: Calendario de trabajo (fuente: Real Academia Española)

Este nivel todavía permite la proliferación y definición insuficiente de los procesos de ingeniería de software. Los proyectos comparten principalmente sus experiencias en materia de administración de proyectos, pero sus métodos técnicos pueden diferir. Aún existe incomunicación entre proyectos, grupos, personal y gerencia.

El nivel 2 identifica prácticas de sentido común que son aplicables en todo tipo de organizaciones de desarrollo de software, independientemente de su rubro, tamaño o ambiente de desarrollo. La ausencia de cualquiera de sus prácticas simplemente pone en peligro el éxito de la empresa.

Dentro de cada nivel de madurez se enfatizan distintos puntos. Estos puntos se redefinen en cada nivel con el nombre de "Key Process Areas" (KPA) o también llamadas áreas clave del proceso. Cada conjunto de áreas clave corresponde a un nivel particular de madurez que tiene que ser completado para poder pasar al siguiente nivel. Cada área tiene una serie de objetivos y actividades para aclarar el alcance.

### **Áreas clave del proceso en el nivel 2.**

Las prácticas clave para este nivel son:

- Administración de subcontrataciones.
- Administración de configuraciones.
- Administración de requerimientos.
- Aseguramiento de calidad.
- Seguimiento de proyectos.
- Planeación de proyectos.

### 2.7.1.3 Nivel 3 o Definido

La empresa ha definido un conjunto de procesos, metodologías y herramientas comunes a todos los proyectos iniciados por la corporación. Los procesos comunes están suficientemente documentados en una biblioteca accesible a todos los desarrolladores. Todo el personal ha recibido el entrenamiento necesario para entender el proceso estándar. Existen pautas y criterios definidos para adaptar dicho proceso a las necesidades y características propias de cada proyecto. El nivel de definición es detallado y completo. La dependencia (o el riesgo de depender) en individuos irremplazables es baja.

En este punto la organización ha logrado todas las metas específicas y genéricas de las áreas del proceso, se caracteriza por el buen entendimiento de los procesos, los procedimientos, herramientas y métodos que son usados por el personal de desarrollo.

La dirección organizacional tiene claro los objetivos a lograr, cada proceso esta perfectamente detallado a un nivel lo suficientemente claro, las dudas no son un factor preocupante ya que las normas y los métodos están dirigidos apropiadamente.

#### **Beneficios de implantar el nivel 3.**

La base de datos que reúne estadísticas de los proyectos pasados y en curso, permite planificar y comparar el rendimiento. Existen mecanismos de comunicación entre proyectos y departamentos, lo que garantiza una visión común del producto y una rápida acción para enfrentar los problemas. Las empresas que se encuentran a este nivel sobresalen por la satisfacción del personal.

En empresas de nivel 1 habitualmente se escuchan quejas y acusaciones. A nivel 3 los empleados tienen una alta valoración de los procesos y entienden claramente la manera en que afectan su desempeño habitual. Los gerentes pueden realizar su verdadera función, que es la de administrar. El hecho de realizar revisiones tempranas en forma regular mejora visiblemente la calidad de los productos y minimiza las reiteraciones innecesarias. Curiosamente muchas organizaciones de nivel 1 realizan revisiones de pares, pero lo hacen de manera inconsistente y al primer signo de pánico las suspenden.

La organización ha definido el proceso como base para una implantación consistente y de mejor entendimiento. Avances tecnológicos son introducidos. Procesos para la administración e ingeniería son documentados, estandarizados e introducidos a nivel organizacional.

Las empresas que se encuentran en este nivel se caracterizan por disponer de un grupo de procesos cuyo objetivo es el de mejorar la calidad durante el desarrollo del software.

Así mismo, en este nivel ya se utiliza una metodología para el desarrollo de software que describe las actividades técnicas y de gestión requeridas para la adecuada ejecución de cada proceso.

Las áreas clave del proceso se mencionan a continuación.

### **Áreas clave del proceso en el nivel 3.**

Las prácticas clave para este nivel son:

- Definición de los procesos de la organización.
- Administración integrada del software.
- Administración de requerimientos.
- Programa de entrenamiento.
- Coordinación entre grupos.
- Ingeniería del producto.
- Revisiones.

#### **2.7.1.4 Nivel 4 o Administrado**

En este nivel la corporación mide la calidad del producto y del proceso de software. Ambos, producto y proceso, son seguidos en forma cuantitativa y se controlan mediante métricas detalladas. La capacidad de rendimiento del proceso es previsible. Las mediciones permiten detectar cuando las variaciones del rendimiento se salen de los rangos aceptables, de manera que se pueden tomar medidas correctivas para asegurar la calidad.

Aquí la organización ha logrado todas las metas específicas de las áreas del proceso asignadas a los niveles de madurez 2, 3, y 4 y las metas genéricas asignadas a los niveles 2 y 3.

Se establecen objetivos cuantitativos para la calidad y actuación del proceso. Los objetivos cuantitativos están basados en las necesidades del cliente, usuarios finales, la organización, e implantadores del proceso. Se entiende calidad y actuación del proceso en las condiciones estadísticas y se manejan a lo largo de la vida del desarrollo del sistema.

Se identifican causas especiales de variación del proceso y se corrigen las fuentes de dichas variaciones para prevenir las ocurrencias futuras.

Una distinción crítica entre la madurez nivel 3 y madurez nivel 4 es la previsibilidad de actuación del proceso. En la madurez nivel 4, la actuación de los procesos están controlados usando las técnicas estadísticas y cuantitativas, y los resultados son predecibles. En el nivel 3, sólo son cualitativamente predecibles.

#### **Beneficios de implantar el nivel 4.**

La empresa es capaz de proponerse metas cuantitativas para la calidad de los productos y de los procesos de software. Es posible medir la productividad y calidad de los mismos en cualquier instante del tiempo.

Los proyectos pueden controlar la variación del rendimiento de sus productos para mantenerla dentro de fronteras cuantitativas aceptables. Es posible discriminar las variaciones significativas en el rendimiento del proceso de la variación (ruido) al azar, particularmente dentro de líneas de productos establecidas.

En este nivel, la organización ha iniciado la medición de sus procesos y el análisis de datos; los procesos de software y de productos son entendidos y controlados cuantitativamente. Información detallada es recolectada generando los grandes cambios en control de calidad que se requieren para asegurar la obtención de productos óptimos.

Es necesario aclarar que el hecho de contar con un sistema de métricas de software no significa que se esté en el nivel 4. Existen varias empresas de nivel 1 que miden cuidadosamente el número de defectos detectados durante las pruebas o tests (no es casualidad que les interese tanto). Es una virtual señal de alarma que les dice cuán graves son sus problemas, pero la inmadurez de sus procesos no les permite hacer nada efectivo, excepto tal vez abortar el producto para evitar un daño mayor que puede resultar de distribuirlo a los clientes.

Son muy raras las empresas que han decidido implementar este nivel. No son muchos especialistas de procesos que realmente tengan experiencia práctica, o incluso que entiendan bien las áreas claves de proceso del nivel 4. Son solamente 2 prácticas, pero imposibles de alcanzar si no se ha implementado firmemente los 2 niveles de madurez anteriores.

#### **Áreas clave del proceso en el nivel 4.**

Las prácticas clave para este nivel son:

- Administración de la calidad del software.
- Administración cuantitativa de procesos.

### 2.7.1.5 Nivel 5 u Optimizado

En este nivel, las organizaciones se encuentran en un proceso de mejora continua. Se usan todos los procesos y técnicas modernas, lo mismo que la administración cuantitativa. Las organizaciones se enfocan en la mejora a través de técnicas y procesos de prevención de defectos, cambios en tecnología y cambios en procesos. Menos del 0.1% de las organizaciones en el mundo se encuentran en este nivel de madurez.

#### **Beneficios de implantar el nivel 5.**

La organización entera se aboca al mejoramiento continuo de cada proceso. La corporación cuenta con los medios para identificar las debilidades y reforzar en forma proactiva el desarrollo, con objeto de prevenir la ocurrencia de defectos. Los datos relativos a la eficacia del proceso de software se usan para analizar el costo y el beneficio de usar nuevas tecnologías y de implantar cambios al proceso de software.

Durante el desarrollo se analizan los defectos para determinar sus causas. Los procesos de software se evalúan para prevenir que los defectos conocidos vuelvan a ocurrir, de esta manera las lecciones aprendidas son difundidas a otros proyectos.

Según estadísticas del SEI, el tiempo promedio para avanzar entre los niveles de madurez es el siguiente:

- De nivel 1 a nivel 2, 23 meses.
- De nivel 2 a nivel 3, 22 meses.
- De nivel 3 a nivel 4, 28 meses.
- De nivel 4 a nivel 5, 17 meses.

Según estadísticas publicadas por el SEI sólo Estados Unidos, Canadá, Japón, Singapur, India, Hong Kong, México y Colombia, poseen industrias de alta madurez en Ingeniería de Software.

Algunas de las empresas que han alcanzado este nivel son TATA Consultancy Services<sup>12</sup>, NeuralSoft<sup>13</sup>, Caja Madrid<sup>14</sup>, IBM<sup>15</sup>, Microsoft<sup>16</sup>, sólo por citar algunas. Las pocas que lo han logrado no divulgan sus secretos para mantener su ventaja competitiva. Este nivel es un estado ideal, una especie de paraíso o nirvana, que probablemente nunca será alcanzado por la mayoría de las empresas productoras de software.

### Áreas clave del proceso en el nivel 5.

Las prácticas clave para este nivel son:

- Gestión del cambio de tecnología.
- Gestión del cambio del proceso.
- Prevención de defectos.

---

<sup>12</sup> Referencia. <http://www.tcs.com/spanish/>

<sup>13</sup> Referencia. <http://www.neuralsoft.com.ar/index.asp>

<sup>14</sup> Referencia. <http://www.cajamadrid.es/CajaMadrid/Home/puente?pagina=0>

<sup>15</sup> Referencia. <http://www.ibm.com>

<sup>16</sup> Referencia. <http://www.microsoft.com>

### 2.7.2 Áreas del proceso

Una área del proceso (conocido en inglés como **process area**), es en sí, un racimo de prácticas relacionadas en una área que, cuando son realizadas colectivamente, satisfacen una serie de metas importantes en el mejoramiento significativo de los procedimientos. Todos los modelos CMMI hacen uso de estas áreas del proceso que como ya se había comentado están organizadas en niveles de madurez.

### 2.7.3 Metas específicas

Las metas específicas (**specific goals**) aplican a un área del proceso y se dirigen únicamente a las características que describen lo que debe llevarse a cabo para satisfacer las necesidades de dicha área. Las metas específicas son requeridas como componentes del modelo y son usadas en las valoraciones que determinan si un área determinada está satisfecha.

### 2.7.4 Prácticas específicas

Una práctica específica (**specific practice**) es una actividad que es considerada importante logrando la meta específica asociada. Las prácticas específicas describen las actividades con las cuales se espera producir resultados dentro de un área del proceso.

### 2.7.5 Metas genéricas

Las metas genéricas (**generic goals**) son llamadas así porque la misma declaración de la meta aparece en las múltiples áreas del proceso. Cada área del proceso tiene sólo una meta genérica. El logro de una meta genérica en una área del proceso significa el control mejorado y bien planificado sobre el manejo de los procesos asociados con dicha área, así estos procesos son probables de ser eficaces, repetibles, y duraderos. Las metas genéricas son requeridas dentro del modelo para determinar mediante apreciación si un área del proceso esta cubierta o no.

### 2.7.6 Prácticas genéricas

Las prácticas genéricas proporcionan institucionalización<sup>17</sup> para asegurar que los procesos asociados con las áreas del proceso serán eficaces, repetibles, y duraderos. Las prácticas genéricas están catalogadas por las metas genéricas y características comunes.

### 2.7.7 Características de la Institucionalización.

La Institucionalización es un aspecto crítico de mejoramiento de procesos y es un concepto importante dentro de cada nivel de madurez. Cuando se menciona más adelante en las descripciones de los niveles, la Institucionalización implica que el proceso esta arraigado en la manera en la que el trabajo es realizado.

Un proceso controlado se institucionaliza realizando lo siguiente:

- Repasando las actividades, el estado de las mismas, y los resultados del proceso en niveles directivos, y tomando la acción correspondiente.
- Controlando el proceso usando las técnicas cuantitativas estadísticas y otras tales como calidad del producto y calidad de servicio.
- Supervisando y controlando la actuación del proceso contra los planes por realizar y tomar las acciones correctivas.
- Proporcionando los recursos adecuados (incluso el fondo, las personas, y herramientas).
- Poniendo los productos de trabajo designados bajo los niveles apropiados de dirección.
- Asignando responsabilidad y autoridad por realizar el proceso.
- Evaluando el proceso, sus productos de trabajo, y sus servicios.
- Entrenando a las personas realizando y apoyando el proceso.
- Siguiendo planes establecidos y descripciones del proceso.
- Agregándolo a las políticas organizacionales.

---

<sup>17</sup> Institucionalización: Su significado en este contexto aplica a que dichas prácticas serán establecidas como algo que debe realizarse al menos para garantizar los procedimientos que deriven de su aplicación.

## 2.8 Características comunes

Las prácticas que deben ser realizadas por cada área clave del proceso están organizadas en cuatro *características comunes*, las cuales constituyen atributos que indican si la implementación y la institucionalización de un proceso clave es efectivo, repetible y duradero. Las prácticas identificadas en la característica común de las actividades realizadas describen qué se debe implementar para establecer una capacidad de proceso.

Cuatro características comunes organizan las prácticas genéricas de cada área del proceso, dichas características (conocidas dentro del modelo como **common features**) son componentes ejemplares que no están clasificados de forma alguna. Son sólo agrupaciones que proporcionan una manera de presentar las prácticas genéricas. Cada característica común se designa por una abreviación como se muestra a continuación:

- **Compromiso de realización (Commitment to Perform)**

El compromiso de realización describe las acciones que la organización debe tomar para asegurar que el proceso se establezca y sea permanente. Normalmente abarcan el establecimiento de políticas y liderazgo a nivel de la organización.

- **Habilidad de realización (Ability to Perform)**

La capacidad de realización describe las condiciones previas que deben existir en el proyecto o la organización para poder aplicar el proceso de software en forma efectiva. Normalmente enfocan los recursos, las estructuras de la organización, el entrenamiento y condiciones en general para iniciar un procedimiento específico.

- **Dirección en la implementación (Directing Implementation)**

La dirección en la implementación describe la necesidad de tener un control absoluto (en la medida de lo posible) sobre los procesos y analizar perfectamente bien los resultados de cada etapa.

- **Verificando la implementación (Verifying Implementation)**

La verificación de la implementación describe los pasos que se deben seguir para asegurar que las actividades se llevan a cabo de acuerdo con el proceso establecido. Básicamente abarca las revisiones y las auditorías efectuadas por la alta gerencia, los jefes de proyecto y el grupo encargado de la garantía de calidad de software.

## **2.9 Prácticas genéricas por característica común**

Como ya se vio, cada característica común pertenece a una meta genérica que como su nombre lo sugiere debe cumplirse por cada una de las áreas del proceso, por lo tanto, para una mejor comprensión de las actividades a realizar, se han dividido en cuatro características que son comunes por cada meta genérica, dentro de estas características existen a su vez prácticas genéricas a realizar.

Es importante mencionar que estas prácticas son deseables, pero en la realidad debido a cuestiones de intereses propios de la empresa o de algún otro índole no se llevan a cabo como deberían, pero se mencionaran a continuación porque se considera que conllevan una importancia especial dentro del proceso de desarrollo de sistemas, se está convencido que si se ejecutan cabalmente pueden ayudar a mejorar enormemente la calidad y las posibilidades de entregar resultados más confiables y permanentes.

### **2.9.1 Compromiso para realizar (práctica genérica)**

#### **❖ 2.9.1.1 Establecer una política organizacional**

Por lo general, el nivel gerencial es el responsable de hacer visible a todas las áreas los principios, dirección y expectativas durante el desarrollo de un sistema o proyecto en general. Comúnmente la política organizacional regula y define la manera en la cual cada uno de los procesos debe ser desarrollado y cuales serán los lineamientos a seguir por los integrantes de un equipo de trabajo.

Esto viene a ser algo así como un protocolo de comunicación que hará efectivas y consistentes cada una de las acciones realizadas para así poder cumplir con cada una de las tareas asignadas en el menor tiempo y máxima calidad posible.

## 2.9.2 La habilidad para realizar (práctica genérica)

### ❖ 2.9.2.1 Planear el proceso

Esta parte es medular dentro de cualquier área de desarrollo, por principio se debe entender que realizar un plan que será llevado por todo un equipo implica una serie de factores que deben ser analizados de una manera “fina”.

Se debe definir un objetivo perfectamente claro y concreto, esto permitirá que cada miembro del equipo de trabajo conozca la meta a alcanzar, una vez entendido esto se procede a definir objetivos más pequeños que puedan ser alcanzados en tiempos realistas.

Es necesario asignar autoridad y responsabilidad a las personas adecuadas, esto se define a nivel directivo ya que esta decisión requiere experiencia y conocimiento amplio del negocio.

El entrenamiento de la gente es necesario a todos los niveles, ya que el éxito de la integración e implementación de un proyecto dependerá en mucho del conocimiento técnico y de recursos disponibles que puedan aprovecharse al máximo.

### ❖ 2.9.2.2 Proveer los recursos

El propósito de esta práctica genérica es asegurar que los recursos necesarios para llevar a cabo un proceso ya definido en un plan estén disponibles cuando se necesiten, esto considera tener a la gente adecuada para el desarrollo, contar con la tecnología que no necesariamente debe ser lo más actual, debe ser óptima y susceptible de ser actualizada en el momento que sea necesario, también es importante contar con todas las herramientas como pueden ser paquetes informáticos preparados para llevar los proyectos con orden y apego a las normas establecidas, tener acceso a toda la información relevante incluyendo objetivos a corto, mediano y largo plazo o cualquier decisión que sea convenida a los niveles gerenciales.

La interpretación del término “adecuado” depende de los factores y posibles cambios durante el desarrollo de un proyecto, cuando algo se genera tal como se planeo o mejor de lo que se esperaba, se dice que es adecuado, pero siempre dependerá en gran medida de la percepción de los niveles directivos que al analizar los resultados den su visto bueno o no sobre el desempeño y calidad de un proceso determinado.

### ❖ 2.9.2.3 Asignar responsabilidades

La gente asignada a un proyecto debe tener la autoridad apropiada para ejecutar las responsabilidades asignadas. Éstas se pueden definir de acuerdo a descripciones detalladas sobre la labor a desempeñar y es válido también asignar responsabilidades dinámicamente, esto se debe a que durante un desarrollo de acuerdo al nivel de dificultad, las responsabilidades deben ser cambiadas a otros miembros del equipo.

Es importante recalcar que la tarea encomendada debe ser entendida por la persona seleccionada, ya que se ha visto en muchas ocasiones que cuando se asigna una responsabilidad determinada a un miembro del equipo, éste muchas veces no comprende la dimensión de lo que enfrenta y los resultados finales no son los acordados y por consiguiente producen retrasos de tiempo y esfuerzo en el resto del grupo.

### ❖ 2.9.2.4 Capacitar al personal

Esta práctica aclara la importancia de contar con personal con las habilidades, experiencia y destreza suficiente para desempeñarse dentro de un proyecto. Muchas veces estas personas son tomadas de la misma empresa y de consultoras externas formando así un equipo de desarrollo integral, pero el nivel de conocimientos sobre las herramientas varía mucho por lo general.

Es por esta razón que la capacitación se vuelve un punto necesario para tener la certeza de que todo el equipo comparte el mismo nivel de conocimientos sobre la aplicación o herramienta a usar, y que se puede definir un plan adecuado en base al dominio del negocio adquirido en este entrenamiento.

Muchas empresas buscan a gente preparada que pueda cumplir con los objetivos establecidos, pero es bien conocido que en la práctica los sistemas cambian constantemente, las necesidades van siendo cada vez más exigentes, y como existe una gran diversidad en el ámbito de desarrollo se debe unificar el conocimiento así como la información entre los miembros del equipo para aprovechar las habilidades que cada uno posee, la capacitación debe ser constante en tiempos bien definidos para incrementar los conocimientos en el personal de trabajo y con esto aumentar la calidad en el desarrollo de los sistemas.

### ❖ 2.9.2.5 Establecer un proceso definido

Existe una gran variedad de herramientas que permiten llevar y ejecutar el desarrollo de un sistema informático de manera controlada, algunas de ellas son usadas para poder definir las tareas a realizar, el monitoreo de tiempos, los niveles de avance, etc.

Pero antes de todo esto, existe algo aún más importante, el establecer un proceso definido, esto implica primero que nada, conocer el objetivo a desarrollar y entenderlo claramente, esto necesita de juntas entre miembros de niveles directivos y gerenciales que decidirán lo qué se debe hacer y el cómo se hará.

Toda información generada será comunicada a los miembros de un equipo de trabajo que tendrán en sus manos la responsabilidad de cumplir con las tareas asignadas en los tiempos definidos en dichas reuniones.

Cada tarea será un proceso a desarrollar, se debe definir lo que dicho proceso tendrá que hacer, de lo contrario será un esfuerzo improductivo. Para poder realizar esta función existen herramientas que ayudan a definir los niveles de responsabilidad, el total de tareas a realizar, la complejidad y los tiempos promedio para desarrollarlas considerando la etapa de análisis, diseño y construcción.

Es importante mencionar que las empresas por lo general establecen los procesos a realizar pero los tiempos de entrega de un producto son limitados, si no se conforma un proceso bien definido que permita indicar la ruta a seguir para cumplir con el objetivo, muy probablemente se tengan desperdicios de recursos, tiempo y dinero, que en sí, es lo que más afecta a una institución. Por ello se debe garantizar mediante el uso de procesos bien conformados que se pueda entregar un producto en tiempo y forma, aunada por supuesto la calidad que debe mantenerse o superarse por cada proyecto que se va realizando.

### 2.9.3 Dirigiendo la aplicación (práctica genérica)

#### ❖ 2.9.3.1 Identificar e involucrar a las personas clave

Dentro del desarrollo de un sistema informático no se puede hacer a un lado a la gente que debe estar involucrada dentro del mismo, esto implica elegir al responsable o responsables que pueden ser desde un director de área hasta el nivel más bajo de la pirámide, cuando un sistema se desarrolla o se modifica requiere especial atención, ya que por lo regular la complejidad llega a ser bastante elevada, es por esta razón que todos aquellos involucrados en el proyecto deben estar presentes durante la planeación, toma de decisiones, coordinación o resolución de problemas e interactuar constantemente en cualquier situación que se presente, ya que siempre surgen situaciones adversas que durante una planeación pudieron no haber sido consideradas y que tendrán que ser resueltas en caso de presentarse por las personas clave que puedan determinar la causa de alguna situación fuera de plan así como su posible solución.

#### ❖ 2.9.3.2 Monitorear y controlar los procesos

Dentro del desarrollo de un proceso, el monitoreo se vuelve una función realmente crucial para el éxito o fracaso de un proyecto, en esta etapa el nivel gerencial es el responsable de seguir muy de cerca las actividades realizadas por el equipo de trabajo, esto se debe a que se entiende que a este nivel se comprende mejor el negocio y cuales son los puntos medulares que no se pueden descuidar.

Este monitoreo se realiza todos los días, se deben verificar los resultados, hacer análisis del progreso, de los tiempos, realizar actividades preventivas y correctivas, medir los resultados, evaluar e identificar los problemas que se van presentando y que en algunos casos se repiten con frecuencia.

Esta actividad implica un gran esfuerzo por parte de todo el equipo de trabajo, ya que este monitoreo ayuda a controlar el progreso o en su defecto el retraso de un proceso y darle solución oportuna de manera que no afecte al resto de los procesos en desarrollo, es por eso, que mientras más control se tiene de algo es más fácil poderlo ajustar a las necesidades de la empresa y sacarlo adelante.

### ❖ 2.9.3.3 Recopilar información para el mejoramiento de los procesos

Cuando en un proyecto el desarrollo de los procesos va siendo una labor cada vez más rutinaria, el aprendizaje así como los resultados van siendo cada vez más óptimos, esto permite llevar un registro de la forma correcta en la que se hacen las cosas, el mayor propósito de este proceso es recopilar la mayor cantidad de información que haga más fácil las tareas de desarrollo a los miembros del equipo. Esta actividad por lo general se realiza generando un repositorio que no es más que un sistema que concentra toda la información relevante que debe ser consultada por todo el equipo y actualizada constantemente por los mismos integrantes para que todos tengan acceso al mismo conocimiento para así minimizar los tiempos y los costos en las etapas de desarrollo.

## 2.9.4 Verificando la aplicación (práctica genérica)

### ❖ 2.9.4.1 Evaluación objetiva

La evaluación de un proceso implica primero que nada que haya credibilidad entre los resultados obtenidos y los esperados, que los niveles directivos y gerenciales puedan comprobar la efectividad y calidad de un proceso ya realizado, por lo general siempre se esperan resultados favorables, aunque muchas veces los procesos no se realizan conforme al plan por varias razones, en ese momento entra la experiencia y la evaluación objetiva para determinar si un proceso ha sido realizado conforme al plan o sólo cumple con las mínimas necesidades. En esta etapa será realmente importante tomar decisiones rápidas y efectivas para garantizar que los procesos se vean mejorados o en su defecto tratar de ajustarlos lo más posible a los lineamientos principales establecidos en principio por los altos niveles directivos dentro de la organización.

### ❖ 2.9.4.2 Análisis del estado del proceso en la alta gerencia

El análisis se vuelve la actividad más importante antes, durante y después del desarrollo de un proceso, la alta gerencia debe tomar la responsabilidad de vigilar, monitorear y proveer de todo lo necesario al equipo de trabajo así como definir el estado en el que se encuentra un proceso de acuerdo a su visión y análisis de los resultados.

## 2.10 Características de un proceso maduro

Para distinguir un proceso maduro de uno que no lo es, resulta necesario conocer las diferencias entre ambos, por lo tanto, según la experiencia se sabe que un *proceso inmaduro es fundamentalmente personal, no está documentado, es difícil compartirlo con otros miembros del equipo, no es fácil reproducirlo en nuevos proyectos, no hay entrenamiento, no todo el mundo lo conoce, no se mide, se aplica a veces solamente, es percibido como poco eficiente, es interpretado de manera distinta entre los miembros de un mismo equipo, etc.*

En una organización de software inmadura los procesos son generalmente improvisados por el personal y sus administradores durante el curso del proyecto. Aún si un proceso de software ha sido especificado, no es rigurosamente seguido. En la organización de software inmadura se espera a que las cosas sucedan para reaccionar, es por esto que los administradores se encuentran enfocados en resolver las crisis inmediatas. Planes y presupuestos son rutinariamente excedidos porque los mismos se encuentran basados en estimaciones irrealistas. Cuando a estas organizaciones le son impuestas fechas límites para la entrega del producto, la funcionalidad y la calidad son generalmente comprometidas para cumplir con los tiempos.

Por el contrario, una organización de software madura posee una amplia habilidad para la administración del desarrollo de software y el mantenimiento de procesos. El proceso de software es correctamente comunicado al personal existente y a los nuevos empleados, las actividades de trabajo son realizadas de acuerdo a lo planeado. Los procesos definidos son actualizados cuando es necesario, y las mejoras en el desarrollo son controladas a través de pruebas pilotos y/o análisis de costo-beneficio. Los roles y responsabilidades en los procesos definidos son claros a través del proyecto y de la organización.

En una organización madura, los administradores monitorean la calidad de los productos de software y los procesos que los producen. Existe un objetivo que establece las bases cualitativas con las cuales se analizarán la calidad del producto y los procesos. Planes y presupuestos están basados en estadísticas históricas y son realistas; los resultados esperados para costos, tiempos, funcionalidad y calidad del producto son generalmente alcanzados. En general, una disciplina de proceso es continuamente seguida por todo el personal que entiende el valor de hacerlo, y la infraestructura necesaria para soportar los procesos existentes.

Por lo tanto, en la tabla 2.1 se define lo que se consideran como *características de un proceso maduro*, esto en base a la experiencia e investigación realizada en diferentes empresas las cuales manifiestan su necesidad de poder al menos tener un parámetro que les permita saber en tiempo y forma si los procesos que realizan van encaminados al éxito o al fracaso.

Un proceso puede considerarse maduro si cumple con los siguientes criterios:

CRITERIO	DESCRIPCION
<i>Está definido</i>	<i>El proceso es claro, sistemático y suficientemente detallado. Además existe acuerdo entre el personal, la gerencia y los proyectos respecto al proceso que se va a utilizar.</i>
<i>Está documentado</i>	<i>Esta escrito en un procedimiento publicado, aprobado y fácilmente accesible. Una de las mejores maneras es a través de una Intranet para apoyar los proyectos de desarrollo (ver página detallada).</i>
<i>El personal ha sido entrenado en el proceso</i>	<i>Los ingenieros de software y la gerencia han recibido cursos y entrenamiento en cada proceso que aplica a su trabajo.</i>
<i>Es practicado</i>	<i>El proceso definido debe ser usado en las tareas habituales llevadas a cabo por los proyectos. El entrenamiento y la adaptación del proceso a la realidad de la empresa debieran garantizar su aplicación en la vida real.</i>
<i>Es apoyado</i>	<i>La gerencia no sólo debe firmar y promover los procesos definidos, sino que además debe asignar responsabilidad al personal y al los jefes de proyecto por su cumplimiento.</i>
<i>Es mantenido</i>	<i>El proceso es revisado regularmente, para asegurarse que está adaptado para satisfacer las necesidades reales de los proyectos.</i>
<i>Está controlado</i>	<i>Los cambios y puestas al día del proceso son revisados, aprobados y comunicados oportunamente a todos los usuarios.</i>
<i>Se verifica</i>	<i>La gerencia mantiene mecanismos para asegurarse que todos los proyectos siguen el proceso vigente.</i>
<i>Se valida</i>	<i>Se asegura que el proceso mantiene concordancia con los requerimientos y estándares aplicables.</i>
<i>Se mide</i>	<i>La utilización, los beneficios y el rendimiento resultante del proceso se miden regularmente.</i>
<i>Puede mejorarse</i>	<i>Existen mecanismos y apoyo de la gerencia para revisar e introducir cambios en el proceso, de manera que se pueda mejorar su eficacia e incorporar nuevas metodologías.</i>

Tabla 2.1 Características de un proceso maduro

Como se ha podido notar hasta este punto, la calidad del producto y de los procesos van de la mano, cuando la calidad no se incluye en un sistema, el proceso de desarrollo se encarga solo de encontrar y componer los defectos. Como resultado, los proyectos se enfocan a reparar los defectos conforme a su grado de importancia para los usuarios. Las personas que no tienen una cultura de calidad causan retrasos en las entregas de los proyectos, porque si los incluyen, el costo del proyecto se elevará, seguramente estos individuos salen más baratos, pero los errores cometidos son realmente caros.

En el caso de los modelos de gestión y aseguramiento de la calidad, aún es poca la difusión de los mismos en la comunidad dedicada a esta área, y se necesita no sólo conocer las perspectivas internacionales de la calidad, sino también informar de las consecuencias que podría ocasionar al implementar alguno de los modelos.

Las organizaciones que desarrollan software en México son en su mayoría pequeños y microempresarios que no se interesan por utilizar métodos de control de sus proyectos, su perspectiva radica en cumplir sólo con los requerimientos en las fechas estipuladas, sin ofrecer a los clientes servicios de valor agregado. La necesidad de crear e implementar un proceso continuo para la administración efectiva incluye a todos los miembros (proveedores, clientes, desarrolladores).

Hasta este punto se ha hablado de la importancia que tiene la calidad en los procesos de desarrollo y su implicación para que cualquier organización se mantenga compitiendo en el mercado internacional que hoy día es absolutamente estricto y no perdona a los competidores débiles. Así mismo, ya se ha hablado del modelo CMMI como base para alcanzar estabilidad, calidad y óptimo desarrollo en los procesos de software.

En nuestro país existe una profunda ignorancia sobre la manera correcta de implementar el modelo CMMI o acoplarlo, de manera tal que sirva como una herramienta que permita a cada miembro de un equipo de trabajo llegar a cada objetivo de manera satisfactoria.

En el siguiente capítulo se muestra como aplicar dicho modelo de la manera más sencilla posible y totalmente acoplada a nuestra idiosincrasia, actualmente se sabe que muchas empresas sobre todo de carácter financiero, por la naturaleza de su giro, están adoptando modelos, estrategias y metodologías que les permiten ofrecer servicios de mayor calidad y así minimizar al máximo el riesgo de entregar productos de bajo nivel y de alto costo.

# Capítulo 3



Las prácticas claves como ya se vio describen lo que es necesario hacer pero no definen el cómo hacerlo. Las premisas básicas son:

- La calidad de un producto está determinada por la calidad del proceso que se usa para producirlo.
- La calidad y la productividad no pueden separarse.

El CMMI no es una “bala de plata” y no encauza todos los puntos importantes para un proyecto exitoso. Por ejemplo, este modelo no dirige actualmente la experiencia en aplicaciones de dominio particular, abogando por tecnologías de software específicos, o sugiriendo como seleccionar, contratar, motivar y retener personal competente, aunque estos puntos son cruciales para el éxito de los proyectos.

Las organizaciones de software pueden tener 10 años o más para construir el fundamento para una cultura orientada hacia un proceso de mejoramiento continuo. Aunque a lo largo de la década, los programas de mejoramiento de procesos son ajenos a la mayoría de las compañías, este nivel de esfuerzo es requerido para producir organizaciones de madurez de software.

## Objetivo

Proporcionar los elementos básicos (procesos, técnicas, métricas...) que permitan planificar y controlar eficazmente proyectos de desarrollo de software, para lograr los objetivos de tiempo y calidad de los mismos.

### 3.1 Modelo Metodológico de Desarrollo de Soluciones

En la figura 3.1 se muestra de forma gráfica la metodología de desarrollo de sistemas propuesta:

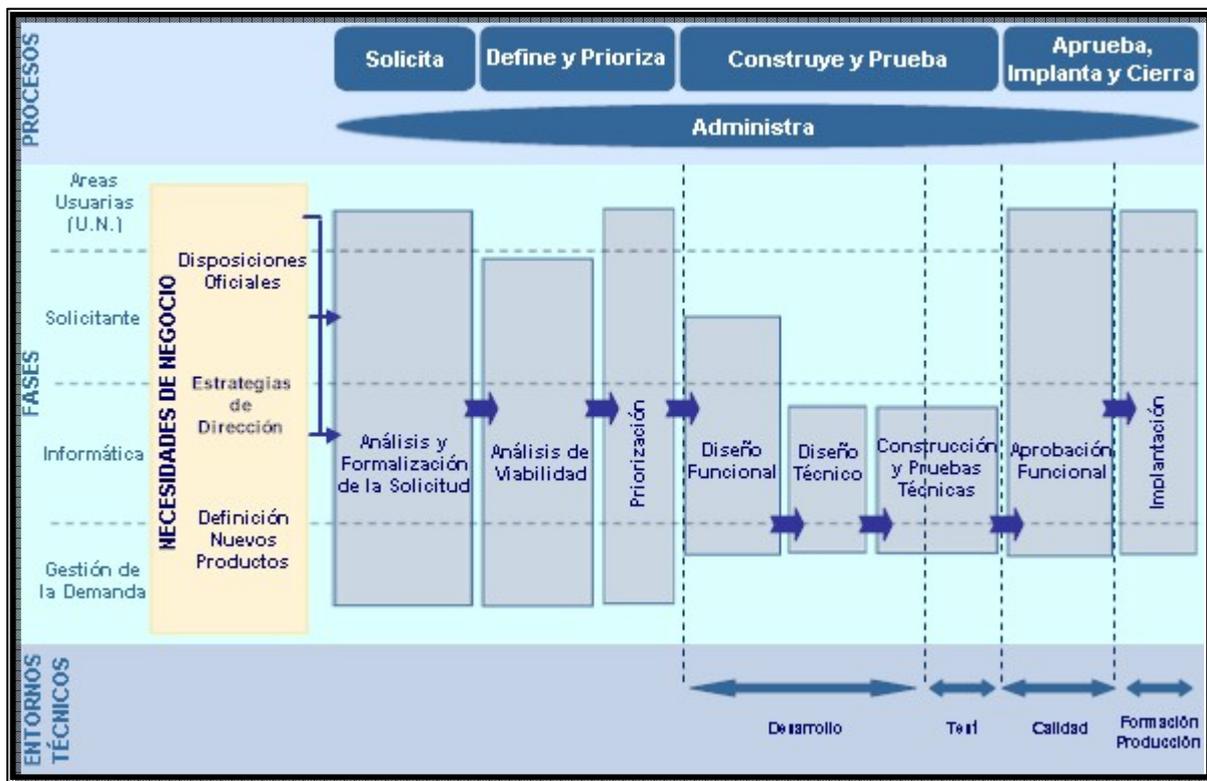


Figura 3.1 Metodología en el desarrollo de soluciones

#### Proceso 1: SOLICITA

**Objetivo:** El área solicitante describe a detalle los trabajos requeridos por entes externos (organismos oficiales y clientes) o internos (área de medios y área de negocio) mediante la elaboración de un escrito y según el caso, de documentos de especificación funcional. Se proporciona al área de Gestión de la Demanda la solicitud formal y la información requerida.

#### Fase 1.1: Análisis y formalización de la solicitud

**Objetivo:** Documentar y registrar la petición del negocio para generar soluciones a un proyecto o servicio generado por las estrategias de la Dirección, cambios o disposiciones oficiales, optimizaciones o labores de mantenimiento (ver figura 3.2).

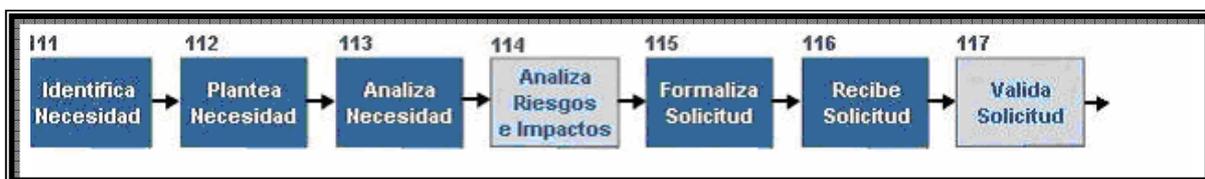


Figura 3.2 Actividades de la fase 1.1

**Proceso 2: DEFINE Y PRIORIZA**

**Objetivo:** Este proceso identifica las necesidades de negocio que pretende cubrir la solicitud planteando un modelo conceptual, estimando costos y priorizando la solicitud con base en las necesidades del negocio y en las capacidades instaladas (ver figura 3.3).

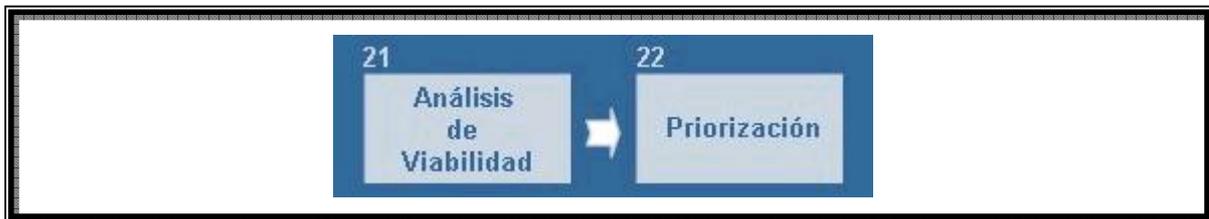


Figura 3.3 Fases del proceso 2

**Fase 2.1: Análisis de viabilidad**

**Objetivo:** Identificar las necesidades del negocio que pretende cubrir la solicitud (petición), planteando para ello una solución conceptual y generando su justificación (ver figura 3.4).



Figura 3.4 Actividades de la fase 2.1

**Fase 2.2: Priorización<sup>18</sup>**

**Objetivo:** Definir el orden de atención del atraso existente, mediante la evaluación del impacto de cada solicitud y la capacidad productiva. Presentar a los diferentes comités la solución conceptual, justificación y evaluación técnico-económica de los proyectos con el fin de obtener su aprobación (ver figura 3.5).

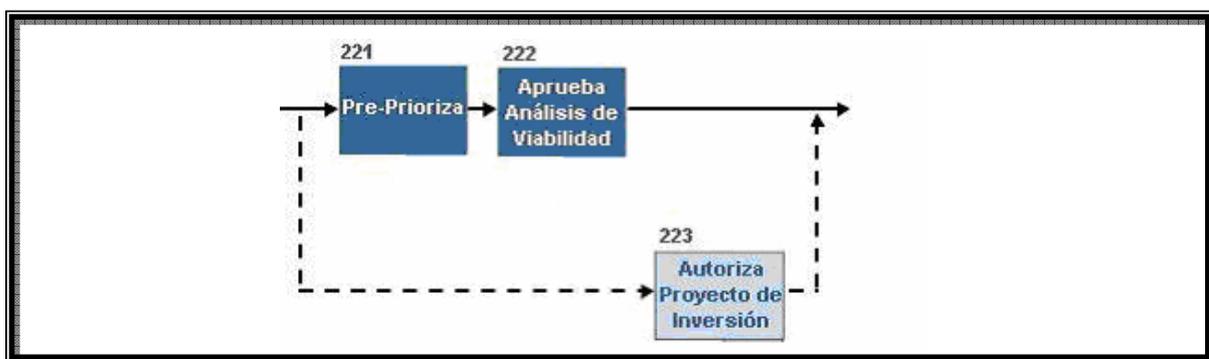


Figura 3.5 Actividades de la fase 2.2

<sup>18</sup> Priorización: término usado coloquialmente por la gente de sistemas para dar prioridad a cada uno de los procesos.

**Proceso 3: CONSTRUYE Y PRUEBA**

**Objetivo:** Las actividades incluidas en este proceso están encaminadas a diseñar funcional y técnicamente la solución, construirla y probarla (ver figura 3.6).



Figura 3.6 Fases del proceso 3

**Fase 3.1: Diseño Funcional**

**Objetivo:** Interpretar las necesidades del negocio como requerimientos (requisitos) funcionales, estructurales y de infraestructura, detallando la solución conceptual en modelos de datos, funciones y de eventos (ver figura 3.7).



Figura 3.7 Actividades de la fase 3.1

**Fase 3.2: Diseño Técnico**

**Objetivo:** Detallar los eventos, funciones y datos en especificaciones técnicas, evaluando los recursos involucrados y en la misma aplicación (ver figura 3.8).

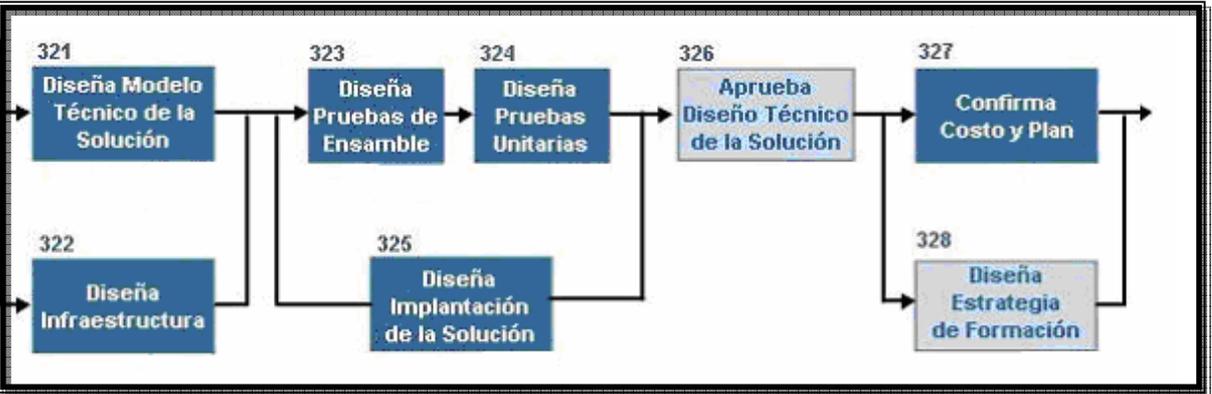


Figura 3.8 Actividades de la fase 3.2

### Fase 3.3: Construcción y Pruebas Técnicas

**Objetivo:** Preparar los entornos, construir la solución y los componentes para su instalación. Confirmar en entornos de prueba, el comportamiento del sistema y de la aplicación bajo distintas categorías y criterios de aceptación (ver figura 3.9).



Figura 3.9 Actividades de la fase 3.3

### Proceso 4: APRUEBA, IMPLANTA Y CIERRA

**Objetivo:** Proceso que inicia cuando la solicitud está lista para ser probada por el área solicitante, todos los componentes deberán ser instalados en el entorno donde se realizarán las pruebas. Una vez que se obtienen las aprobaciones técnica y funcional se procede a la formación e implantación de la solución (ver figura 3.10).

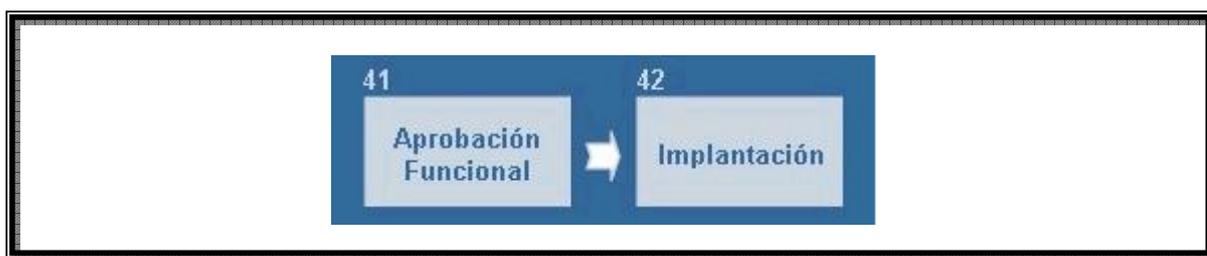


Figura 3.10 Fases del proceso 4

### Fase 4.1: Aprobación Funcional

**Objetivo:** Preparar el entorno correspondiente para llevar a cabo las pruebas de funcionalidad para obtener la aprobación (ver figura 3.11).



Figura 3.11 Actividades de la fase 4.1

### Fase 4.2: Implantación

**Objetivo:** Concluir los trámites de instalación y liberación, llevar a cabo la formación y efectuar una evaluación post-instalación de la solución implementada (ver figura 3.12).



Figura 3.12 Actividades de la fase 4.2

### Proceso 5: ADMINISTRA

**Objetivo:** Guiar a los equipos de trabajo de manera estructurada y flexible para alcanzar resultados a tiempo y dentro del presupuesto autorizado. A través de la administración se propone llevar a cabo la planeación de los trabajos a fin de implementar medidores de desempeño, estableciendo objetivos y resultados tangibles. Apoya la gestión de la solicitud, llevando a cabo la administración de los recursos, comunicación, riesgos, control de presupuestos, entrega en tiempo y ejecución de la solución.

**NOTA.**

*Este proceso se presenta a lo largo de todas las fases ya mencionadas; y no se generan productos nuevos, sólo se modifican productos ya existentes.*

### 3.2 Proceso de Administración de Proyectos

En la figura 3.13 se define estructuralmente el proceso de administración de proyectos:

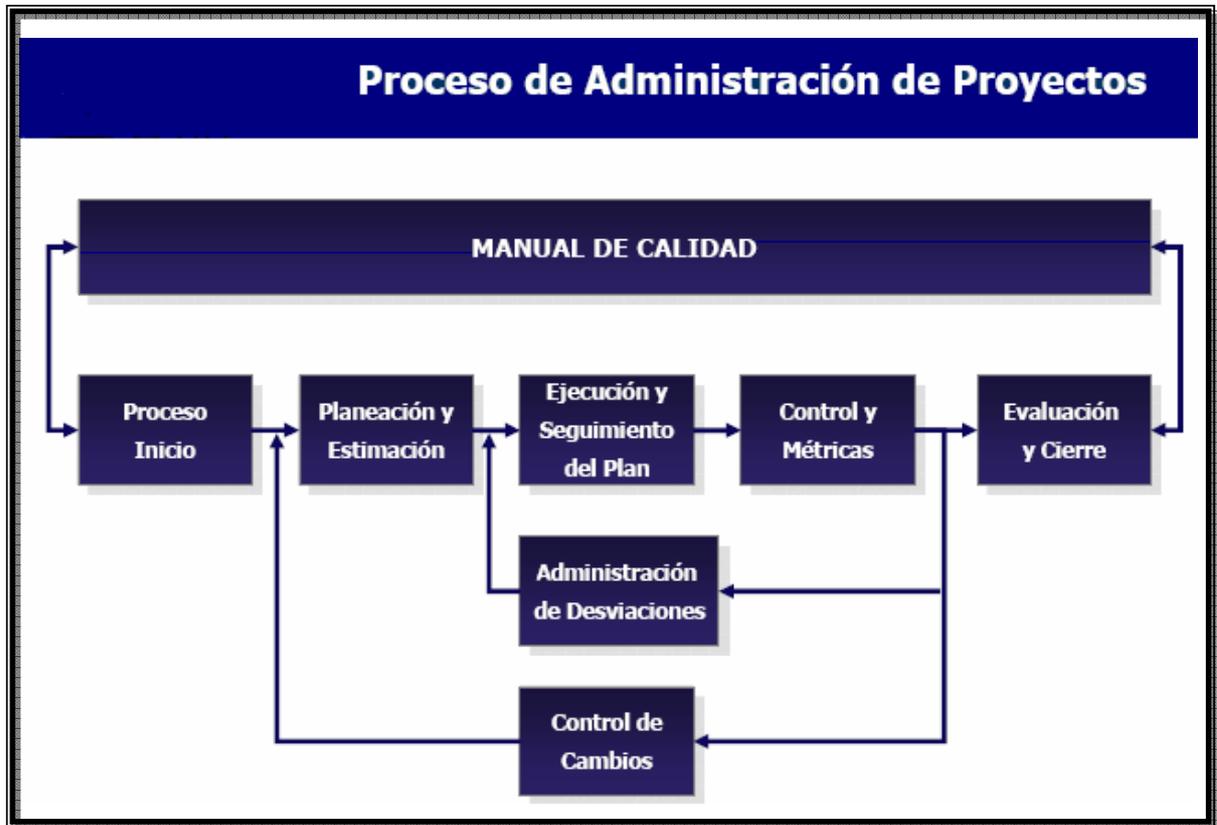


Figura 3.13 Proceso de administración de proyectos

#### 3.2.1 ¿Qué es un Proyecto?

Un Proyecto es un conjunto único de tareas interrelacionadas y bien definidas para lograr resultados establecidos, bajo parámetros de tiempo, costo y calidad.

##### Características de un proyecto:

- ✓ Tiene un inicio y por supuesto un fin determinado.
- ✓ Se conoce el alcance y los objetivos específicos.
- ✓ Se desarrollan productos concretos.
- ✓ Por lo general es complejo.
- ✓ Dinámico.

### **Factores críticos de un proyecto.**

#### **Factores que pueden ocasionar el fracaso de un proyecto:**

- ✓ Objetivo y alcance mal definidos.
- ✓ Seguimiento y control deficiente.
- ✓ Ausencia o mala planificación.
- ✓ Expectativas mal definidas.

#### **Factores clave de éxito dentro de un proyecto:**

- ✓ Soporte y compromiso de todos los involucrados.
- ✓ Clara definición y acuerdo de alcance y objetivos.
- ✓ Riesgos identificados y pasos para minimizarlos.
- ✓ Buena planeación y control.
- ✓ Equipo de trabajo sólido.

#### **Características del grupo de trabajo:**

- ✓ Con las habilidades y conocimientos requeridos.
- ✓ Roles y responsabilidades claramente definidos.
- ✓ Comunicación efectiva.

#### **Características de un Líder de Proyecto:**

- ✓ Conocimientos técnicos y del negocio que se maneja.
- ✓ Comunicación efectiva y actitud de servicio.
- ✓ Correcta administración de proyectos.
- ✓ Buen uso de manuales de calidad.
- ✓ Capacidad de negociación.
- ✓ Manejo de la diversidad.
- ✓ Visión y compromiso.

### 3.2.2 Gestión de la Calidad

La gestión de calidad facilita el desarrollo y entrega de soluciones que satisfagan las expectativas del cliente basado en el principio “first-time-right” o dicho en otras palabras hacer desde el principio *lo apropiado*, de *la forma correcta*, para obtener *el resultado adecuado* y apoyado por modelos, relación con clientes, proveedores, administración de personal y prácticas.

Pero que significan estas tres frases en concreto:

❖ *Lo apropiado*

Definir claramente expectativas, factores claves de éxito y responsabilidades para cumplirlos.

❖ *La forma correcta*

Aplicar las metodologías, procesos, estándares, técnicas, herramientas apropiadas, asignar al equipo de trabajo adecuado y asegurar la continua comunicación e interacción con el cliente.

❖ *El resultado adecuado*

Que las soluciones entregadas contribuyan al logro de los objetivos del cliente, en el tiempo pactado y con el presupuesto asignado.

### 3.2.3 Riesgos

Un riesgo es un evento externo con un grado de probabilidad que ocurra e impacte al proyecto. Por ello es conveniente prevenir los factores que pueden poner en una situación de la cual no se pueda más que corregir lo que está mal, no es lo más óptimo y se sugiere no llegar a este punto.

La figura 3.14 que se muestra a continuación ejemplifica mejor lo antes mencionado:

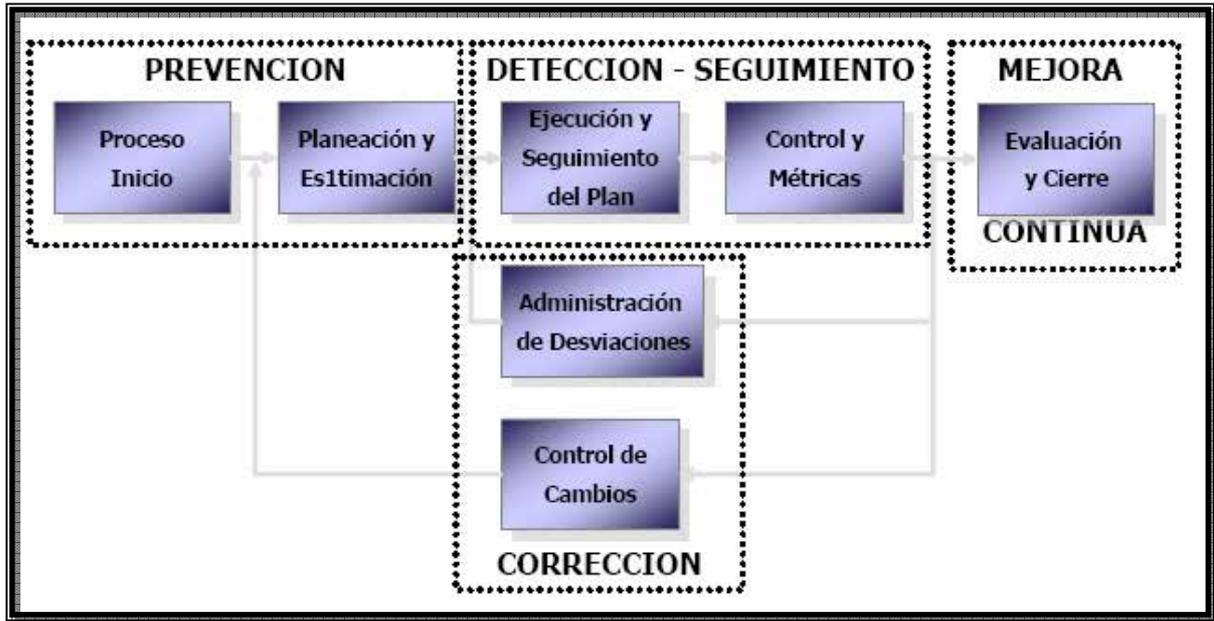


Figura 3.14 Gestión de riesgos

### 3.2.4 Comunicación

La comunicación es una mezcla dinámica de información y ruido. Una vez que la información es recibida, una respuesta positiva o negativa (feedback) debe ser enviada.

La comunicación es el arte de expresar e intercambiar ideas en forma oral o escrita.

#### Factores críticos de éxito.

- Debe difundir misiones, funciones, objetivos, prioridades, riesgos, avances y logros.
- La información se debe difundir a tiempo (cuando es necesaria).
- Se debe de identificar el tipo de información que se recibe.
- Debe haber retroalimentación de la información recibida.
- La información se debe compartir y no ocultar.
- Debe fluir hacia dentro y hacia fuera del área.
- Debe separarse la información y el ruido.
- Debe existir comunicación continua.

### 3.2.5 Proceso de inicio

En la figura 3.15 se observa el proceso de inicio dentro de un proyecto:

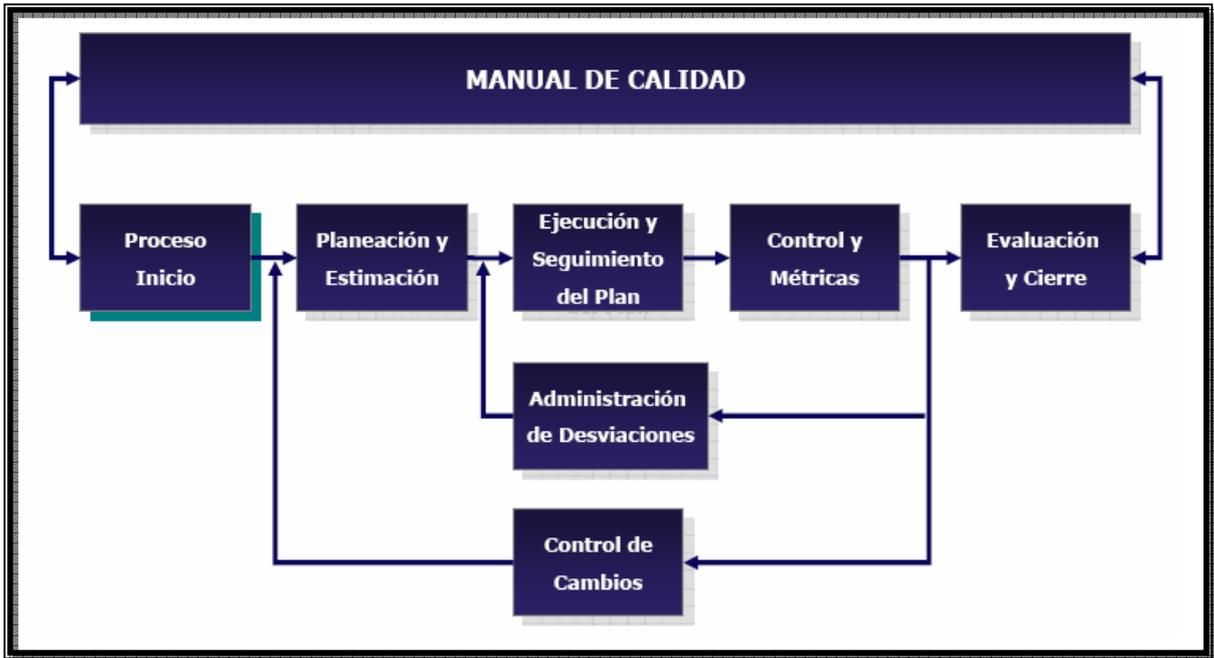


Figura 3.15 Proceso inicio

Es el proceso por el cual se identifican expectativas de cambio, se define el alcance y el impacto; se identifican riesgos y se definen las formas de prevenirlos o de corregirlos en caso de que se presenten.

**Expectativas:** Son las metas esperadas en el nuevo proyecto, esto puede deberse a necesidades de crecimiento, legales, cambio operativo, competencia, rendimiento, etc.

**Alcance:** Es la descripción detallada de los entregables del proyecto. Define el resultado esperado, marca la relación entre los entregables, los objetivos del negocio y los factores de éxito.

#### Factores a tomar en cuenta.

- Dependencias (áreas, aplicativos, proyectos).
- Funciones del negocio a implementar.
- Beneficios del negocio a conseguir.
- Valor entregado.

La tabla 3.1 sintetiza las actividades a realizar en el proceso de inicio:

TAREA	FASE DE DESARROLLO
Revisar y analizar las expectativas, de acuerdo a las solicitudes registradas por el solicitante, priorizar y detallar el alcance de cada una para definir su atención mediante una versión.	<p>Diseño Conceptual</p> <p>Diseño Funcional</p>
Definir y documentar objetivo y alcance del proyecto	
Identificar, documentar y evaluar riesgos, determinando acciones de prevención, medición y corrección	
Definir y realizar acuerdos con las áreas involucradas en el proyecto	
Definir y conformar el equipo de trabajo, identificando perfiles, roles, experiencia y capacidades de los integrantes	
Identificar los factores críticos de éxito del proyecto	
Identificar y evaluar a los proveedores que apoyaran al proyecto	
Establecer los canales de comunicación con el solicitante, proveedores, equipo de trabajo y la dirección para comunicar la participación de cada uno de ellos	

Tabla 3.1 Actividades a realizar en el proceso de inicio

**Factores críticos de éxito.**

- Entender claramente los beneficios del negocio.
- Establecer medidas basadas en resultados.
- Crear una cultura receptiva del cambio.
- Adoptar comunicación efectiva.
- Asegurar un liderazgo efectivo.

### 3.2.6 Planeación y Estimación

La fase de planificación es predeterminar una secuencia de eventos o tareas (ver figura 3.16).

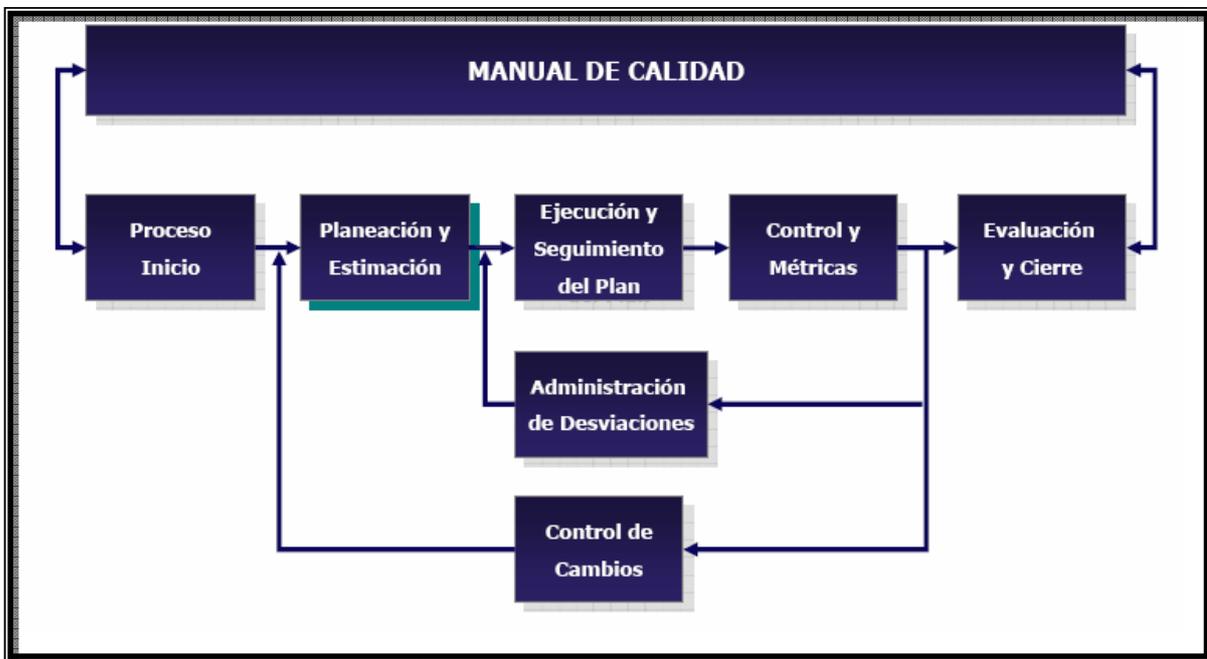


Figura 3.16 Planeación y estimación

Estimación es una interpretación del esfuerzo requerido para completar el alcance definido (ver figura 3.17).

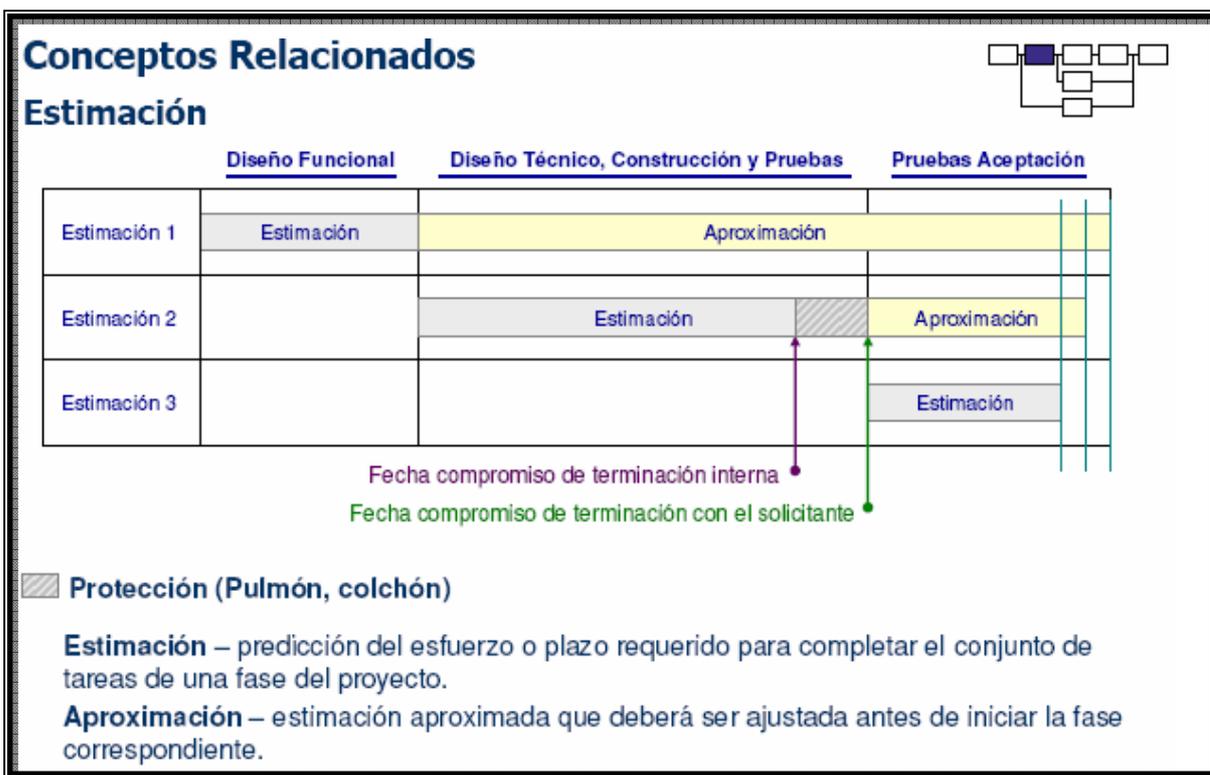


Figura 3.17 Conceptos relacionados

### **Esfuerzo.**

Es la cantidad de trabajo necesaria para completar una tarea o proyecto. El esfuerzo es medido en la cantidad de unidades de trabajo requeridas para completar una actividad (por ejemplo horas/hombre). La palabra esfuerzo, como tal, implica **trabajo** y **costo**. El esfuerzo total esta basado en:

- Esfuerzo para crear cada entregable.
- Tipo y número de entregables.

### **Estimando.**

- Revisar que las tareas están incluidas en el alcance del proyecto.
- Identificar oportunidades de ahorro:
  - Combinación/Separación.
  - Reutilización.
- Grado de complejidad.
- Dependencias de tareas y recursos.
- Nivelación de recursos.
- Considerar riesgos.

### **Riesgo y Prevención.**

Durante el proceso de inicio se detectan los posibles riesgos y se definen las formas de prevenirlos o de corregirlos en caso de que se presenten. Durante la planeación deben revisarse nuevamente y en su caso actualizarse. Deben considerarse sobre todo en la estimación, como parte de la protección, en caso de que lleguen a presentarse situaciones críticas.

La tabla 3.2 muestra las actividades a realizar en la fase de planeación y estimación:

TAREA	FASE DE DESARROLLO
Identificar tareas	<p>Diseño Conceptual</p> <p>Diseño Funcional</p>
Establecer dependencias entre tareas	
Determinar las fechas de inicio y terminación	
Estimar la duración de cada tarea	
Asignar responsables a cada tarea	
Estimar el esfuerzo en horas de cada tarea	
Registrar recursos y porcentaje de participación	
Evaluar posibles riesgos	
Revisar riesgos identificados	
Identificar nuevos eventos que puedan afectar al éxito del proyecto	
El visto bueno del plan de trabajo por el equipo y las áreas involucradas	

Tabla 3.2 Actividades a realizar en la fase de planeación y estimación

**Factores críticos de éxito.**

- Si cambia la fecha de finalización, cambiar el plan de trabajo.
- Incluir los tiempos de administración en el plan de trabajo.
- Aplicar experiencias adquiridas en proyectos anteriores.
- Negociación del plan (alcance, tiempo, recursos, costos).
- Considerar que el líder no va hacer todas las tareas.
- Ser realista, no vivir en un “mundo ideal”.
- Evitar caer en el síndrome del estudiante.
- Conocimiento del trabajo a realizar.
- Documentación clara y detallada.

### 3.2.7 Ejecución y seguimiento del plan

La fase de ejecución y seguimiento es el proceso mediante el cual se lleva a cabo el plan del proyecto, con el fin de desarrollar, mantener y obtener el visto bueno de los entregables, de acuerdo a los tiempos y recursos estimados (ver figura 3.18).

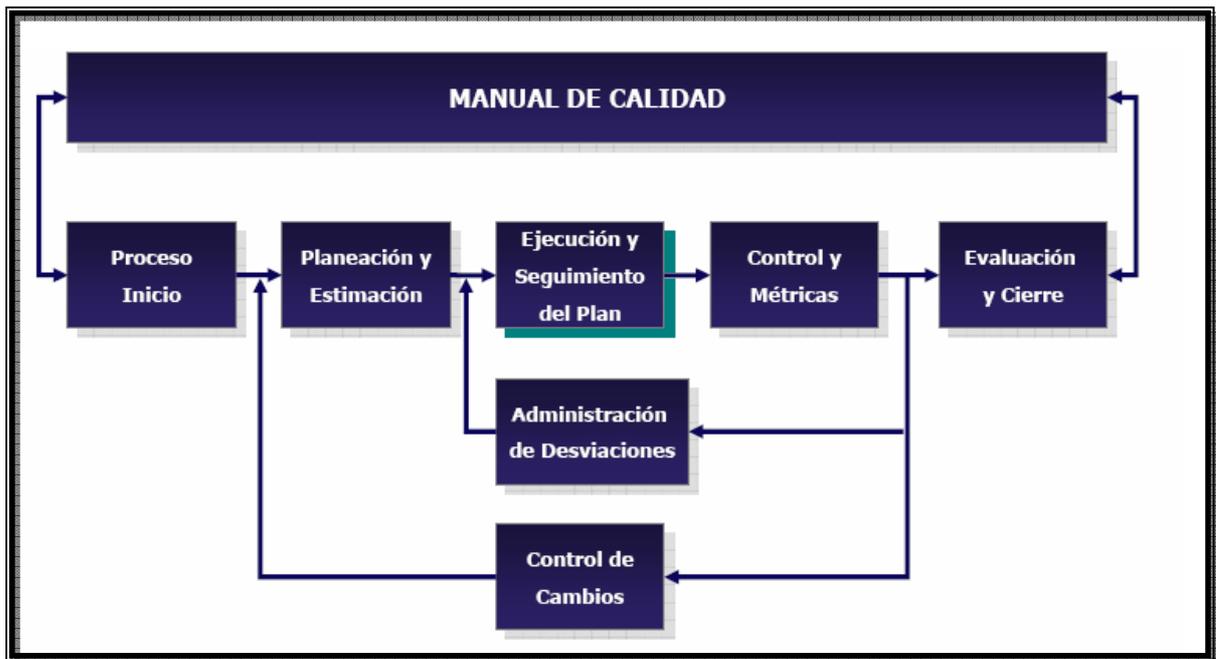


Figura 3.18 Ejecución y seguimiento del plan

#### Riesgos y detección.

Durante la ejecución del plan, se debe:

- Llevar a cabo las acciones pertinentes para corregir los que se hayan presentado.
- Monitorear los momentos en los que se prevé se puedan presentar los riesgos.
- Llevar a cabo las actividades para prevenir los riesgos.
- Detectar nuevos riesgos no considerados.

#### Control del proyecto.

- Reunión de seguimiento con todos los miembros del equipo incluyendo la alta dirección.
- Que la actualización de información sea de preferencia el último día hábil de la semana.
- Debe existir una fecha de corte y respetar los tiempos acordados de entrega.
- Se recomienda realizar reporte de actividades de forma diaria.

La tabla 3.3 muestra las actividades a realizar en la fase de ejecución y seguimiento:

<b>TAREA</b>	<b>FASE DE DESARROLLO</b>
Realizar el trabajo de acuerdo a lo estipulado en plan para cada una de las áreas involucradas	Diseño Funcional  Diseño Técnico  Construcción  Pruebas del Centro de Desarrollo  Pruebas de Aceptación
Actualizar las tareas terminadas en el plan con horas y tiempos reales	
Monitorear los riesgos y actualizar el estado	
Obtener aceptación de los entregables intermedios	
Actualizar semanalmente el avance de las tareas del plan antes de la fecha de corte (todos los viernes de preferencia)	
Monitorear el desempeño de cada integrante del equipo	
Coordinar las dependencias de tareas entre las áreas involucradas	
Realizar juntas de seguimiento con el equipo de trabajo y áreas involucradas	
Llevar a cabo las medidas preventivas de los riesgos	Diseño Conceptual  Diseño Funcional
Llevar a cabo las acciones para la corrección de desviaciones	
Comunicar las desviaciones e impactos	

Tabla 3.3 Actividades a realizar en la fase de ejecución y seguimiento

### 3.2.8 Control y métricas

La fase de control es el proceso de comparar los resultados planificados con los resultados obtenidos y realizar las acciones correctivas necesarias cuando existan diferencias significativas (ver figura 3.19).

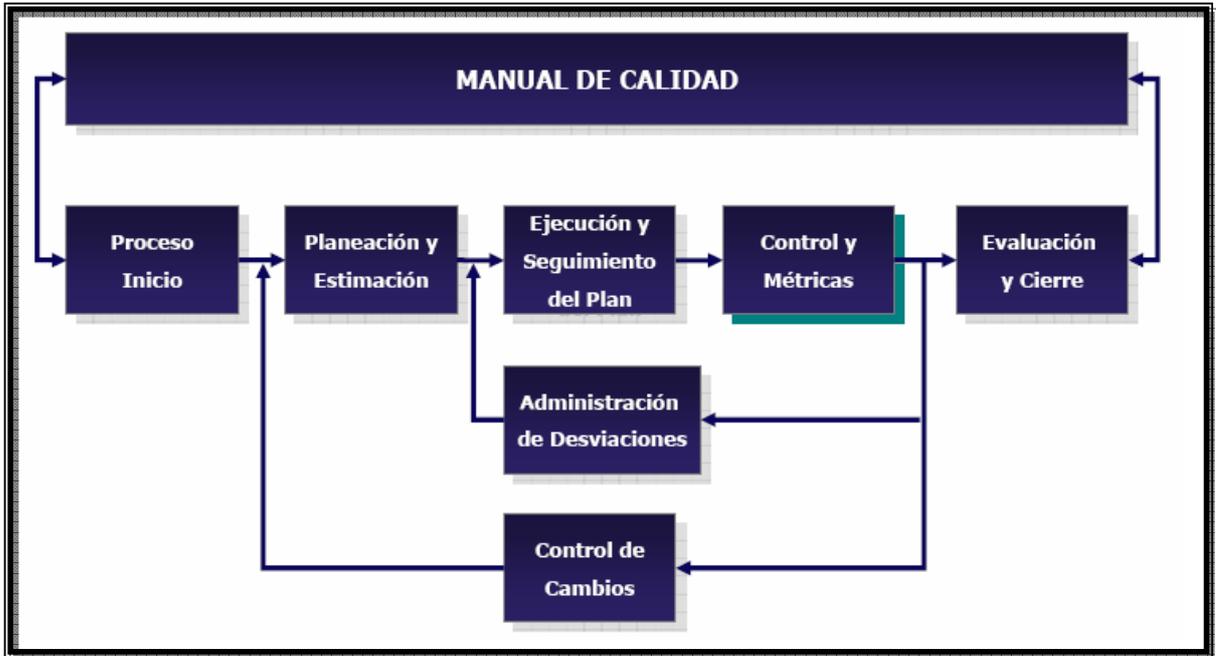


Figura 3.19 Control y métricas

#### Métricas.

- Permiten comparaciones imparciales por cambios en tecnología y procesos.
- Permiten detectar desviaciones del plan y las estimaciones.
- Generan preguntas respecto al objetivo del negocio.
- Ayuda a gestionar el proyecto hacia una mejora.

#### Métricas en Proyectos.

- Tiempo:
  - Línea del tiempo (ver figura 3.20).
  - Curvas "S" de componentes (ver figura 3.21).
- Costo:
  - Gráfico de horas (ver figura 3.22).
- Calidad:
  - Curva "S" de incidencias (ver figura 3.23).

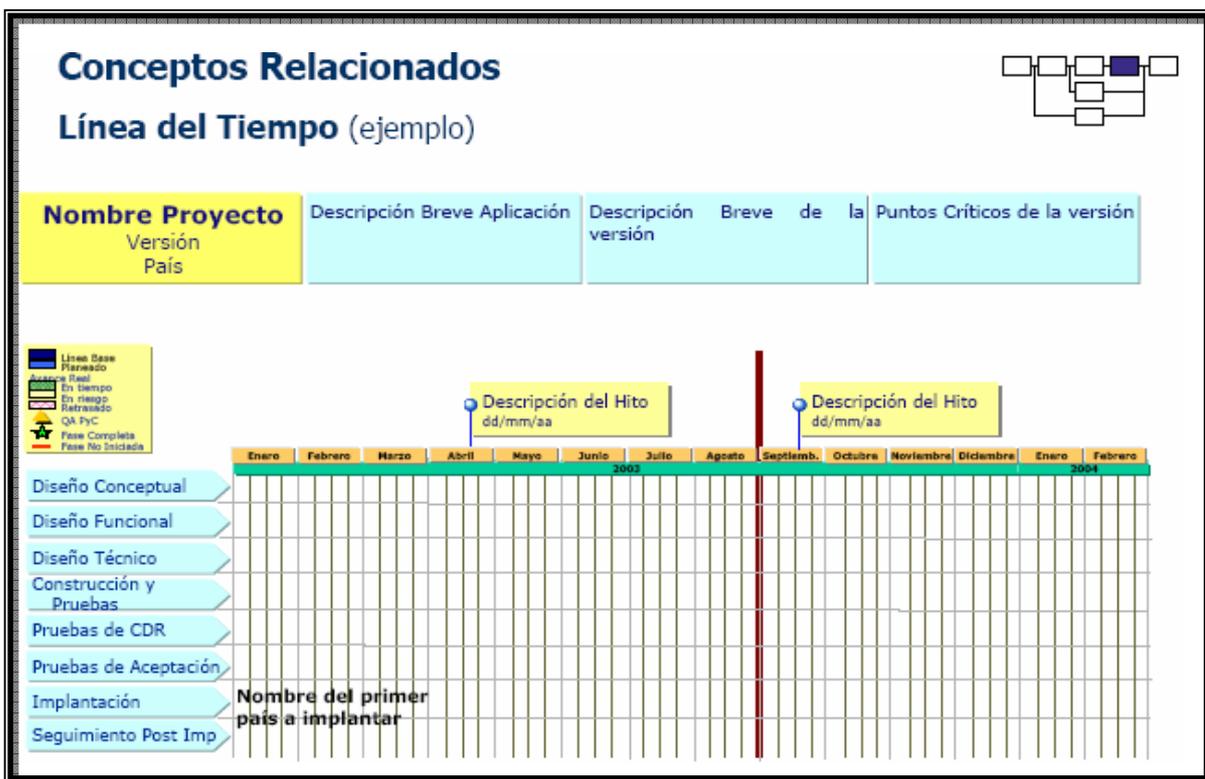


Figura 3.20 Línea del tiempo (ejemplo)

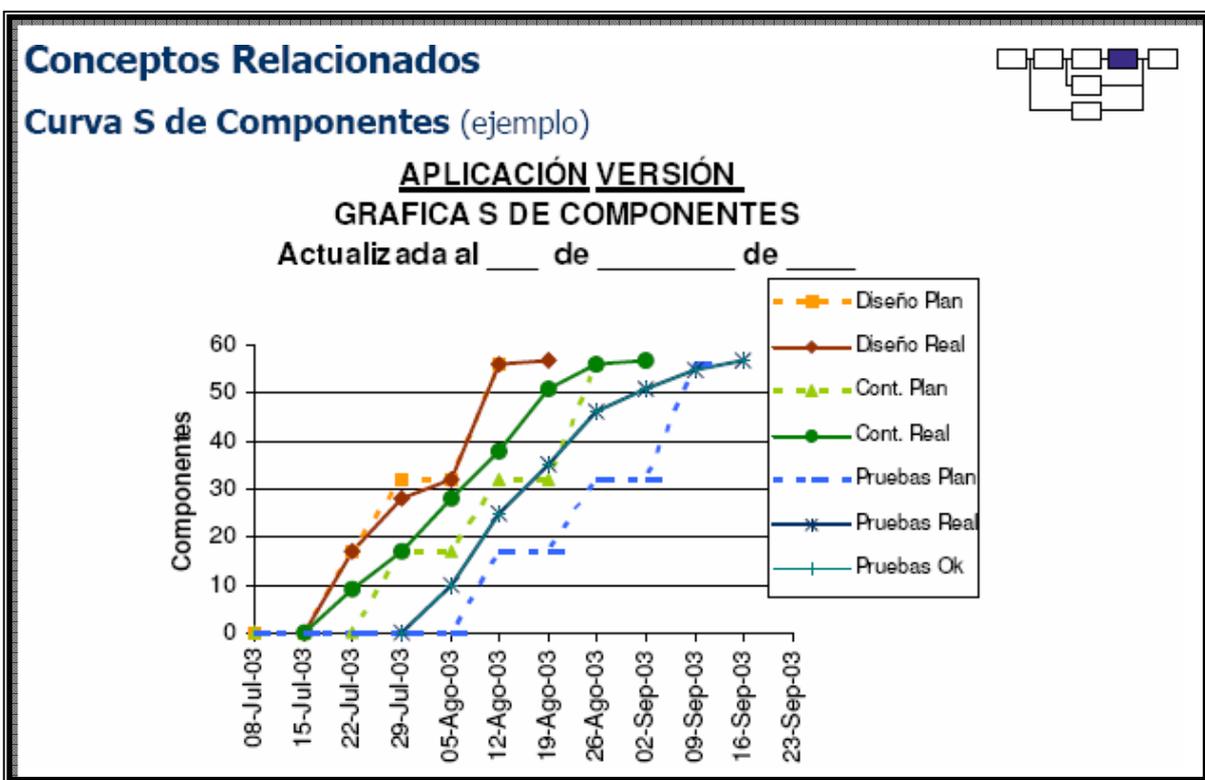


Figura 3.21 Curva S de componentes (ejemplo)

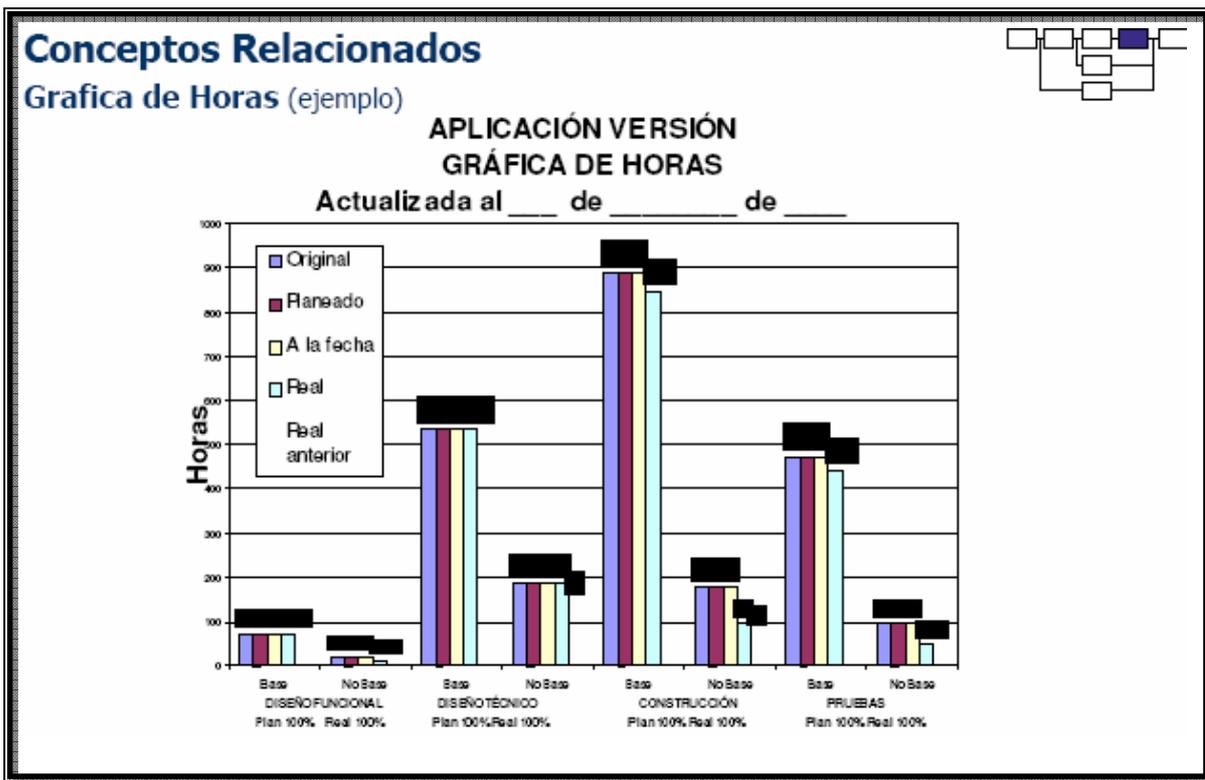


Figura 3.22 Gráfica de horas (ejemplo)

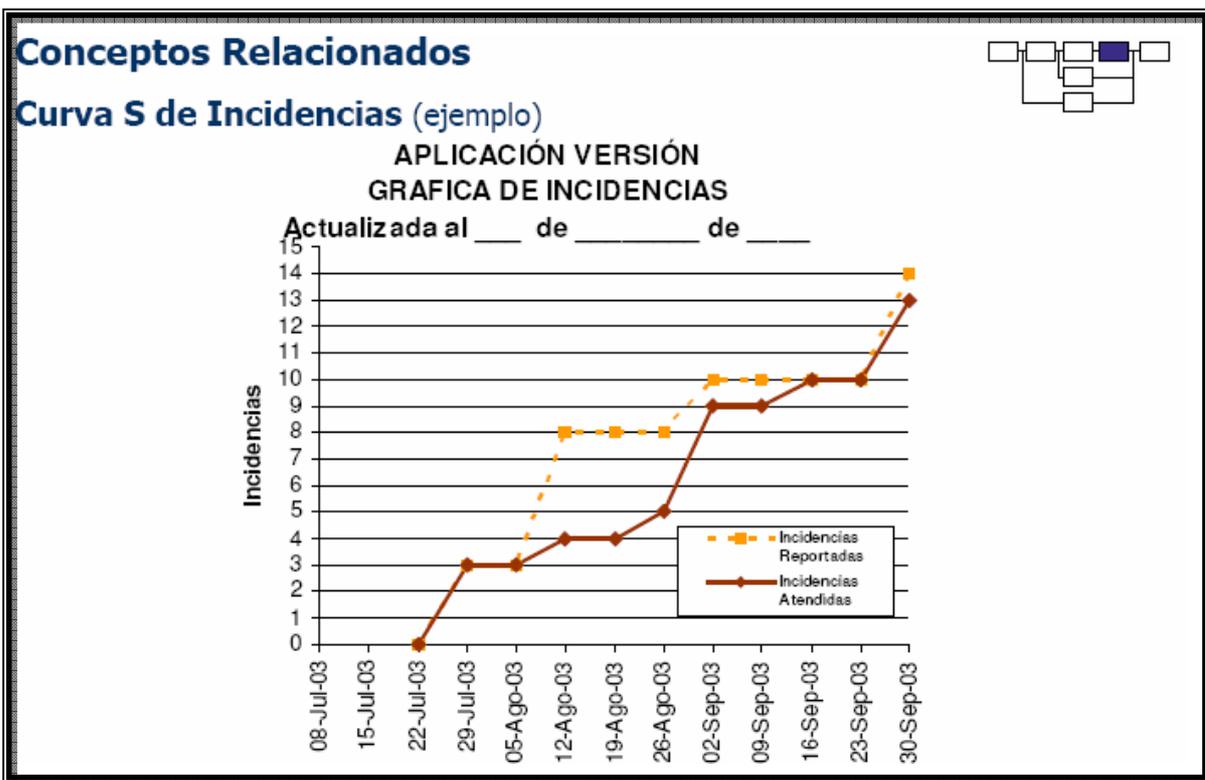


Figura 3.23 Curva S de incidencias (ejemplo)

**Riesgos y detección.**

Durante el control, se debe llevar un monitoreo de los riesgos de acuerdo a:

- Riesgos que ya no se presentaron por la fase en la que se encuentra el proyecto.
- Riesgos que se pueden presentar en la fase actual o en las fases siguientes.
- Riesgos que ya se presentaron y sus consecuencias.
- Medidas de prevención de los riesgos.

La tabla 3.4 muestra las actividades a realizar para esta fase de control y métricas:

<b>TAREA</b>	<b>FASE DE DESARROLLO</b>
<p>Cada semana revisar, analizar, validar con el responsable e imprimir la información del seguimiento del proyecto ya actualizada. Preparar el reporte ejecutivo para la reunión de seguimiento con la Dirección.</p>	<p>Diseño Funcional</p> <p>Diseño Técnico</p> <p>Construcción</p> <p>Pruebas del Centro de Desarrollo</p> <p>Pruebas de Aceptación</p>
<p>Generar las métricas del proyecto (línea del tiempo, gráficas de Productividad, Costo, Calidad)</p>	
<p>Asegurar que el alcance no haya cambiado</p>	
<p>Evaluar impactos derivados de las desviaciones</p>	
<p>Evaluar si se cuenta con recursos suficientes para continuar con el proyecto</p>	
<p>Identificar temas que requieran solución</p>	
<p>Evaluar el estado de los riesgos y su tratamiento en caso de haberse detectado</p>	

Tabla 3.4 Actividades a realizar en la fase de control y métricas

**Factores críticos de éxito.**

- Evaluar el status de los riesgos y su tratamiento en caso de haberse presentado.
- Llevar a cabo una comunicación continua de los avances del proyecto.
- Llevar a cabo reuniones semanales para revisar el status del proyecto.
- Verificar que el proyecto se lleve de acuerdo a estándares.
- Verificar que el alcance del proyecto no ha sido rebasado.
- Evaluar constantemente los recursos para evitar retrasos.
- Identificar oportunamente temas que requieran solución.
- Analizar las métricas obtenidas del proyecto.
- Evaluar la causa de las desviaciones.

### 3.2.9 Administración de desviaciones

Es la óptima gestión de la separación del plan original en cuanto a tiempo, tareas, recursos humanos o materiales y/o costos que sufre un proyecto, ya sea por un evento interno o externo (ver figura 3.24).

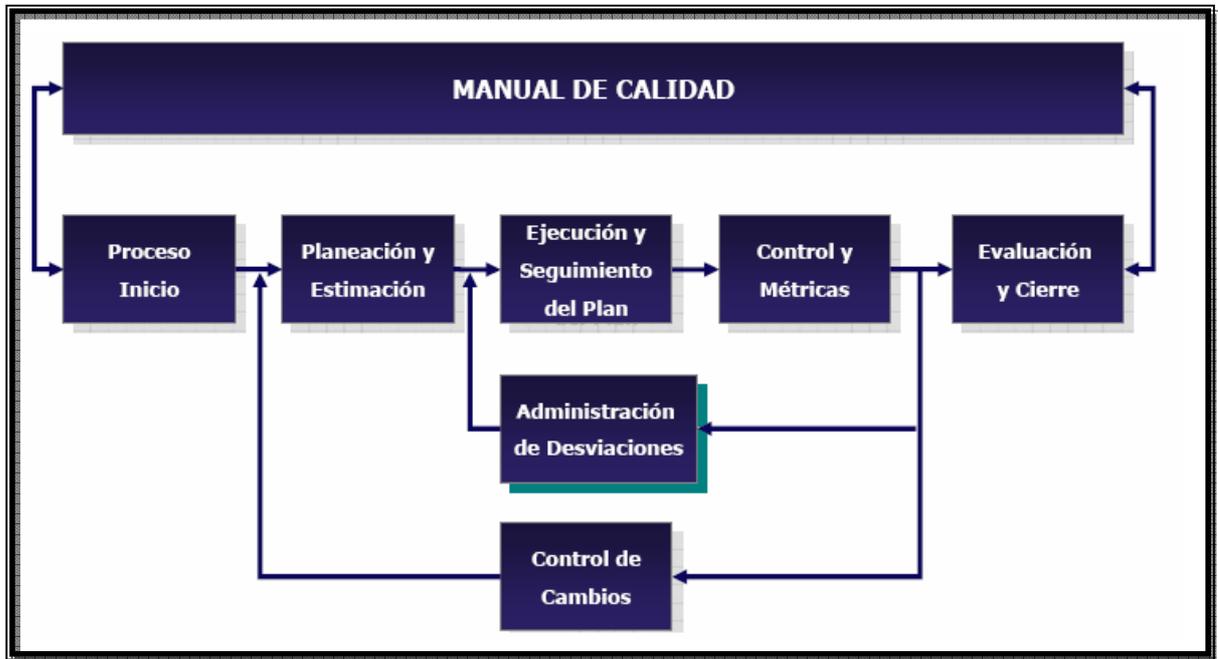


Figura 3.24 Administración de desviaciones

#### Origen de las desviaciones.

- Factores externos (entornos, equipo, medio ambiente, enfermedades).
- Carencia de habilidades, conocimiento, experiencia y/o motivación.
- Demasiadas y/o innecesarias opciones disponibles.
- Tomar atajos no cumpliendo especificaciones.
- Incomprensión del objetivo principal.
- Riesgos (previstos y no previstos).
- Documentación poco clara.
- Malentendidos.

La tabla 3.5 muestra las actividades a realizar para la fase de administración de desviaciones:

<b>TAREA</b>	<b>FASE DE DESARROLLO</b>
Identificar las desviaciones	Diseño Funcional  Diseño Técnico  Construcción  Pruebas del Centro de Desarrollo  Pruebas de Aceptación
Documentar las desviaciones	
Determinar las causas de las desviaciones	
Documentar la importancia e impacto de las desviaciones	
Generar un grupo de acciones que permita corregir las desviaciones	
Obtener aprobación de acciones a tomar	
Evaluar que el alcance no haya sido sobrepasado, y en caso contrario implementar el proceso establecido para el control de cambios	

Tabla 3.5 Actividades a realizar en la fase de administración de desviaciones

**Factores críticos de éxito.**

- La desviación no debe tener gran impacto sobre el proyecto, en estos casos debe ser tratado como cambio.
- En caso de más de una desviación simultánea, priorizar de acuerdo a las tareas pendientes.
- La información de la desviación debe ser clara y detallada, para determinar su causa.
- La desviación debe comunicarse a las áreas involucradas.
- Documentar y administrar las desviaciones presentadas.
- Registrar clara y detalladamente la solución.

### 3.2.10 Control de cambios

Un cambio en el proyecto es la variación de la planeación o estimación de un proyecto originado por la alteración de sus elementos (alcance, tiempo, esfuerzo, riesgo, calidad, recursos) afectando al resto de estos (ver figura 3.25).

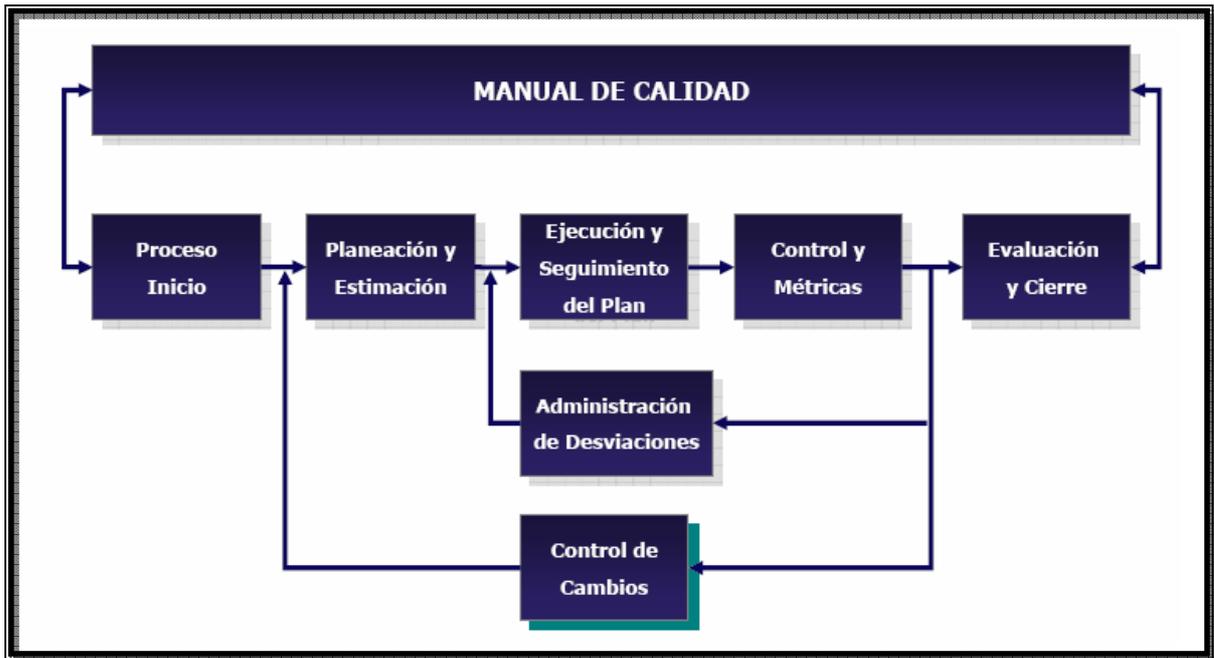


Figura 3.25 Control de cambios

#### Cambios internos.

- Valor agregado, buenas intenciones, sin beneficio real.
- Desviaciones que rebasan el plan.

#### Cambios externos.

- Otros proyectos, prioridades y dependencias.
- Legales y/o gubernamentales.
- Infraestructura.
- Cliente.

La tabla 3.6 muestra las actividades a realizar para la fase de control de cambios:

<b>TAREA</b>	<b>FASE DE DESARROLLO</b>
<b>Evaluar el impacto de cambio de alcance</b>	<b>Diseño Funcional</b>  <b>Diseño Técnico</b>  <b>Construcción</b>  <b>Pruebas del Centro de Desarrollo</b>  <b>Pruebas de Aceptación</b>
<b>Validar la justificación del cambio</b>	
<b>Considerar las alternativas y seleccionar la más adecuada</b>	
<b>Determinar el impacto en tiempo, esfuerzo, riesgo y calidad</b>	
<b>Documentar los requerimientos de cambio y sus implicaciones</b>	
<b>Solicitar el visto bueno de los cambios a la dirección</b>	
<b>Comunicar a la Dirección, equipo y áreas involucradas el cambio</b>	

Tabla 3.6 Actividades a realizar en la fase de control de cambios

**Factores críticos de éxito.**

- Detectar cuales cambios deben incluirse como parte del mismo proyecto y cuales pueden esperar la siguiente versión.
- Es necesario, identificar que no se trate de una desviación que pueda ser solucionada fácilmente.
- La información debe ser clara y detallada, para determinar su origen y su impacto.
- Mantener la mente abierta ante el punto de vista del cliente.
- Mantener un registro de los cambios que se presenten.
- Registrar clara y detalladamente el camino a seguir.
- El cambio debe de comunicarse a todas las áreas.

### 3.2.11 Evaluación y cierre

En esta última fase se lleva a cabo la evaluación de resultados una vez que se concluyeron todas las tareas del proyecto (ver figura 3.26).

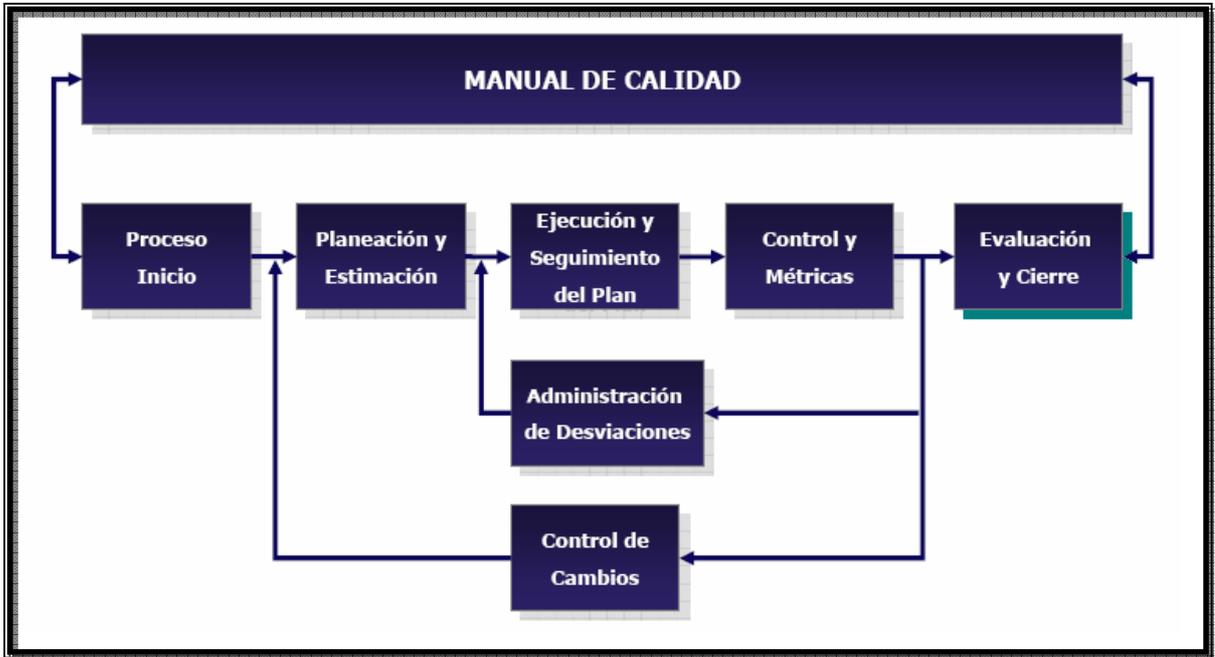


Figura 3.26 Evaluación y cierre

#### Evaluación.

- **Del proyecto:**
  - ¿La comunicación y relación con el solicitante y áreas participantes fue la adecuada?
  - ¿Cuáles fueron los factores de éxito o fracaso del proyecto?
  - ¿Se lograron los objetivos de tiempo, costo y calidad?
  - ¿Se cumplieron las expectativas del solicitante?
  - ¿Fue realista el plan de trabajo original?
  
- **Del equipo de trabajo:**
  - ¿Hubo retroalimentación y/o reconocimiento al trabajo realizado?
  - ¿Cómo contribuyó cada integrante del equipo al proyecto?
  - ¿Hubo desarrollo profesional? (aprendizaje)
  - ¿Cómo fue el trabajo en equipo?
  - ¿Fue el adecuado?

- **Mejora continua:**
  - ¿Que áreas de oportunidad se identificaron para mejorar: capacidades, procesos, modelos, estándares, métricas, herramientas...?
  - Retroalimentar el Manual de Calidad:
    - Procesos.
    - Estándares.
    - Productos.
  - Nutrir, proteger, incubar nuevas ideas.
  - Aprovechar experiencia adquirida.
  - Búsqueda de nuevas ideas.
  - Dar pasos adelante.
  - Reciclar.

La tabla 3.7 muestra las actividades a realizar en la fase de evaluación y cierre:

<b>TAREA</b>	<b>FASE DE DESARROLLO</b>
<b>Obtener aceptación formal de entregables</b>	<b>Pruebas de Aceptación</b>
<b>Finalizar documentación</b>	
<b>Transferir responsabilidad de entregables</b>	
<b>Liberar recursos</b>	
<b>Analizar riesgos presentados</b>	
<b>Comunicar la terminación del proyecto</b>	
<b>Evaluar el proyecto</b>	
<b>Evaluar el equipo de trabajo</b>	
<b>Evaluar las oportunidades de mejora continua</b>	
<b>Definir prioridades de mejora continua</b>	

Tabla 3.7 Actividades a realizar en la fase de evaluación y cierre

**Factores críticos de éxito.**

- Cumplir con las actividades de gestión de calidad y retroalimentarlas.
- Evaluar objetivamente el proyecto y el equipo de trabajo.
- Verificar que la documentación sea clara y completa.
- Comunicar adecuadamente el fin del proyecto.

- Identificar los puntos de mejora continua.
- Comunicar el resultado de la evaluación.
- No dejar puntos abiertos o pendientes.

En general durante un proyecto no se debe descuidar los siguientes puntos:

**Los factores básicos del éxito.**

- El plan de trabajo debe ser realista y basado en experiencias documentadas.
- Optimizar constantemente los procesos de desarrollo.
- Definir claramente los objetivos y el alcance.
- Mejorar la administración de los proyectos.
- Manejar claramente los cambios.
- No asumir eventos o productos.
- Vigilar el control de calidad.
- Administrar las incidencias.

**Los factores básicos relacionados con el equipo.**

- Desarrollar capacidades del equipo (conocimientos, habilidades, experiencia, entre otras).
- Planear de forma razonable los tiempos de trabajo y de asueto de cada recurso.
- Evitar personas "indispensables".
- Evitar el alto nivel de rotación.
- Dar capacitación constante.

En el capítulo siguiente se hablará de una propuesta de calidad que involucra cada uno de los temas ya vistos a lo largo de este trabajo de tesis, aunque conceptualmente en muchos países dicha estrategia ya es utilizada, en nuestro país pocas empresas la dominan y la mayor parte por increíble que parezca no tienen claro el potencial que su uso puede ofrecerles en la optimización de sus recursos y en la minimización de los costos preservando la calidad en cada proceso.

# Capítulo 4



El Centro de Desarrollo Regional es la estrategia de calidad aplicada por una institución financiera y esta orientada a desarrollos informáticos de manera conjunta con los países de Latinoamérica donde tiene presencia.

Este documento pretende definir la forma de trabajo para el Centro de Desarrollo Regional, con el propósito de iniciar su utilización a manera de guía, divulgando los estándares a ser aplicados por las áreas que interactúan dentro y fuera del Centro conocido también en otros países como “fábrica de software”.

## Objetivo

El objetivo del Centro es desarrollar aplicaciones para Latinoamérica con altos niveles de calidad, generando ahorros de tiempo y recursos económicos.

Para lograr el objetivo se plantean las siguientes acciones:

- Establecimiento de estándares de desarrollo de aplicaciones.
- Implantación de herramientas de gestión y desarrollo.
- Definición e implantación de modelos y procesos.
- Establecimiento de indicadores y métricas.
- Comunicación efectiva con el cliente.
- Centralización del conocimiento.
- Reutilización de componentes.

## 4.1 Descripción genérica del CDR (Centro Desarrollo Regional)

En la figura 4.1 se esquematiza la conformación del centro de desarrollo regional:

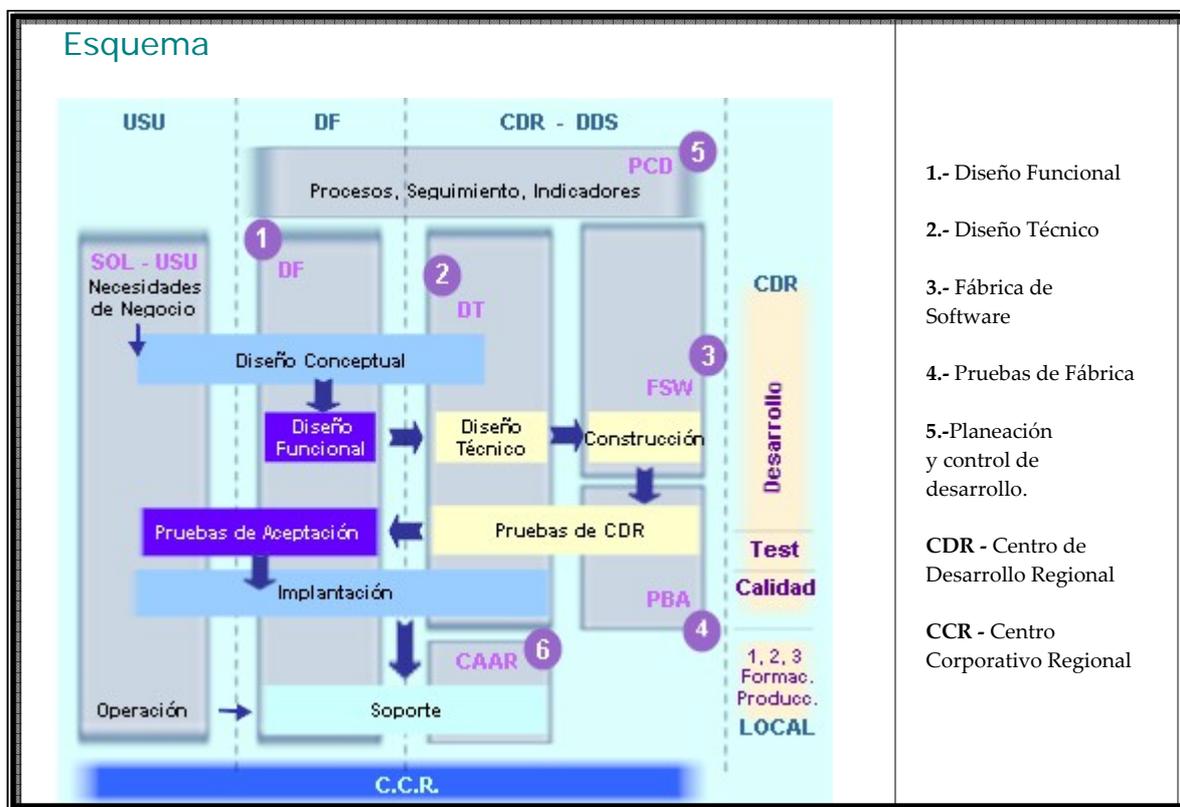


Figura 4.1 Esquema CDR (Centro Desarrollo Regional)

### 4.1.1 Funciones principales

#### Área de Diseño Funcional:

- Elaborar los diseños de las pruebas de aceptación.
- Administrar los objetos funcionales reutilizables.
- Brindar soporte funcional a países (Helpdesk).
- Definir el alcance de la solución requerida.
- Realizar análisis de viabilidad funcional.
- Colaborar en las pruebas de aceptación.
- Apoyo y soporte a la implantación.
- Elaborar el diseño conceptual.
- Elaborar el diseño funcional.

**Área de Diseño Técnico:**

- Elaborar el plan de trabajo global del proyecto.
- Administrar los objetos técnicos reutilizables.
- Elaborar los diseños de las pruebas técnicas.
- Brindar soporte técnico a países (Helpdesk).
- Administrar versiones de aplicaciones.
- Realizar análisis de viabilidad técnica.
- Colaborar en las pruebas técnicas.
- Mantenimiento correctivo.
- Elaborar el diseño técnico.

**Área de Fábrica de Software:**

- Colaborar en el diseño y definición de pruebas.
- Definir la estrategia de pruebas del CDR.
- Verificar el cumplimiento de estándares.
- Ejecutar las pruebas funcionales.
- Controlar el proceso de pruebas.
- Ejecutar las pruebas técnicas.

**Área de Pruebas CDR**

- Colaborar en el diseño y definición de pruebas.
- Definir la estrategia de pruebas del CDR.
- Verificar el cumplimiento de estándares.
- Ejecutar las pruebas funcionales.
- Controlar el proceso de pruebas.
- Ejecutar las pruebas técnicas.

**Área de Planeación y Control del Desarrollo:**

- Dar seguimiento al cumplimiento de los procesos establecidos.
- Coordinar los esfuerzos entre las áreas del CDR y CCR.
- Revisar la aplicación de los estándares establecidos.
- Establecer ANS (Acuerdos de Niveles de Servicio).
- Generar las métricas del cumplimiento de ANS.
- Definir e implantar la operativa de trabajo.
- Administrar los requerimientos del CDR.

### 4.1.2 Estándares guía del área de diseño

*Los principios del Diseño de aplicaciones están basados en:*

- Sencillez, claridad y precisión: generar especificaciones que expongan de manera clara y concisa, sin ambigüedades, como cubrir las necesidades funcionales y técnicas solicitadas.
- Conocimiento, en cuanto a la comunicación, crecimiento y retroalimentación que de lugar a la creación de repositorios de información del diseñador y guías de “checklist”.
- Técnico, en cuanto soluciones / programas con desempeño eficiente, reutilización de código, etc.
- Funcional, en cuanto a la solución propuesta al usuario del sistema.
- Reutilización, reaprovechar el trabajo a distintos niveles.

*Las tareas del Área de Diseño Funcional son:*

- *Definir alcances de las soluciones requeridas;* definición de la meta que deberá ser cubierta por la solución planteada, así como de las condiciones, variables, etc., que afectarán o modificarán el mismo, a través de los medios de comunicación provistos por la metodología:
  - Delimitación de alcance de la solución en la fase preliminar de proyecto.
  - Definición de estrategias de implantación realistas.
  - Análisis preliminar de riesgos e impactos.
- *Elaboración del diseño funcional;* descripción detallada de la funcionalidad a partir de modelos de procesos, de datos y de interacción que cubran la solución, apoyados en los siguientes elementos:
  - Interfases con usuarios, ya sean pantallas, reportes, etc. En este aspecto, un elemento fundamental para garantizar la aprobación funcional será la elaboración de prototipos, en la medida en que aporta una aproximación gráfica a la realidad del método de interacción del usuario con el sistema.
  - *Flujo Operativo:* representación gráfica de los procesos operativos y flujos de información involucrados en la operación diaria del futuro sistema, que ayuden a detectar posibles impactos en la operación de los usuarios.

- **Apoyo a implantación de soluciones y soporte funcional;** en cuanto a participación activa en:
  - Seguimiento post-implantación del sistema y atención de usuarios.
  - Pruebas de aprobación funcional y técnica del sistema.
  - Implantación del sistema en producción.

**Las tareas del área de Diseño Técnico son:**

- **Elaboración del diseño técnico;** descripción técnica de los componentes que forman parte del sistema, a manera de especificar la solución técnica, con el objetivo de:
  - Utilización de flujos de comunicación eficientes con las otras áreas de CDR y CCR: Planeación y Control del Desarrollo, Fábrica de Software, Pruebas de Fábrica, Calidad, Infraestructura y Producción.
  - Documentación implícita a todas las soluciones generadas como soporte de:
    - Futuras labores de mantenimiento y adaptaciones / evoluciones del sistema.
    - Claridad de expresión técnica de las soluciones planteadas.
- **Apoyo a implantación de soluciones y soporte funcional;** en cuanto a participación activa en:
  - Seguimiento post-implantación del sistema y soporte técnico a países.
  - Implantación del sistema en producción.
  - Pruebas técnicas del sistema.

#### 4.1.2.1 Diseño funcional

Este apartado es una síntesis sobre "puntos" y "aspectos básicos" a tener presentes en cada análisis, enfocada a cubrir criterios asociados a los objetivos que se pretenden conseguir en el funcionamiento general de los diferentes grupos de trabajo y su coordinación.

##### **Principios Generales.**

- Generar soluciones con "valor agregado" de las áreas participantes, contemplando:
  - Aspectos comerciales y de negocio.
  - Aspectos administrativos.
  - Aspectos de información.
  - Aspectos de control.
  - Aspectos técnicos.
- Evaluar la posibilidad de mejorar los procedimientos de negocio antes de ser automatizados.
- Generar soluciones integrales, sin caer en contradicciones con otras propuestas.

##### **Conceptos a considerar.**

- Determinar los "puntos críticos", evaluando las deficiencias actuales.
- Conocer o tratar de entender lo "que piensa" el usuario o solicitante.
- Conocer la situación actual en todas sus implicaciones.
- Aportar "soluciones":
  - Que contemplen todos los aspectos relacionados.
  - Que uniformicen procedimientos.
  - Que no necesiten "especialistas".
  - Que se acepten por los usuarios.
- Todo lo que no "entienda" el usuario o solicitante:
  - Producirá incidencias.
  - No lo va a utilizar.

#### 4.1.2.2 Desarrollo del diseño

La documentación del análisis de viabilidad y del diseño funcional deberá contener los siguientes apartados:

Situación actual: Descripción de la situación actual en todos los aspectos (operativos, contables, documentales, flujos de información, etc.)

- Descripción detallada de la solución funcional propuesta a partir de los modelos de: datos, procesos e interacción.
- Puntos Críticos, descripción de los puntos débiles, deficiencias y carencias de la situación actual.
- Evaluación y definición de relaciones con otros procesos de negocio (interfases).
- Descripción global de la estrategia del proyecto.
- Diseño y distribución de informes (reportes):
  - Internos.
  - Externos.
- Descripción global de la solución propuesta.
- Ventajas y/o beneficios:
  - Cualitativos.
  - Cuantitativos.
- Diseño de pantallas.
- Análisis detallado.

#### 4.1.2.3 Pruebas de aceptación

Las pruebas de aceptación contemplan los siguientes puntos:

- El solicitante y el área de diseño funcional realizarán las pruebas de aceptación de la solución desarrollada gestionando los siguientes puntos:
  - Pruebas de la aplicación y verificación de resultados de acuerdo a los diseños.
  - Visto bueno de las pruebas y documentación de ajustes.
  - Definición de responsabilidades durante las pruebas.
- Finalizado el desarrollo informático, es necesario realizar las pruebas funcionales.

#### 4.1.2.4 Concepto de oficina

Para la elaboración del diseño, considerar que la oficina es un punto de venta y de servicio al cliente orientado a conseguir:

- Alta capacidad de gestión y atención al cliente sin que derive en tratamientos administrativos posteriores.
- Cobertura comercial en su gestión de venta de productos.
- Capacidad de incidir en su mercado potencial.
- Mínimos tratamientos administrativos.
- Reducción del "back office"<sup>19</sup>:
  - Centralizar operaciones administrativas.
  - Automatizar procesos operativos.
  - Automatizar controles.

#### 4.1.2.5 Consideraciones al diseñar

Los puntos siguientes son indispensables a la hora de diseñar y se definen por:

##### **Productos.**

- La apertura de productos contempla en su ejecución las operaciones de apertura operativa, primer ingreso, cargo/abono en cuentas personales, solicitudes de talonarios y otras.
- La contratación de productos se realiza dentro del proceso creado para tal fin: pre-venta, contratación, venta cruzada, documentación involucrada.
- Todos los productos tienen filosofía de "cuentas", transacciones operativas de pregunta y respuesta.
- Todos los productos deben ajustarse al proceso de contratación.
- La contratación de productos asocia operaciones relacionadas.
- Atención personalizada al cliente y en un solo punto.

---

<sup>19</sup> Back office: es la parte de las empresas donde se destinan las actividades a gestionar que son propias de la empresa y con las cuales el cliente no necesita contacto directo.

- Los diferentes productos se desarrollan como cuentas de un cliente y en formatos homogéneos.
- Con conocimiento del cliente, productos que utiliza y "perfil" de venta para nuevos productos.
- Manejo de una operación principal que encadena las operaciones asociadas.
- Con controles de entrada que impidan errores / rechazos posteriores.
- Con nomenclatura estándar en la presentación de datos y pantallas.
- Con actualización de bases de datos corporativas en tiempo real.
- Sin requerir documentos preimpresos durante la "contratación".

### **Aplicaciones.**

- El desarrollo transaccional debe contemplar todos los procesos:
  - Control datos → actualización registro → contabilización → control ejecución → acceso a información.
- Preparar las aplicaciones para que todos los datos y las transacciones fluyan en una sola operación al resto de los productos.
- Las entradas de operaciones y movimientos desde transacciones en oficinas son "ejecutivas" y no informativas.
- Evitar "especialistas" para la ejecución de transacciones y resolución de operaciones.
- Cubrir las necesidades del usuario (solicitante) considerando sus expectativas.
- Emplear métodos de corrección de errores uniformes, sencillos y definitivos.
- Preocuparse por el diseño de la información más que por las estructuras.
- Se crean controles de funcionamiento automático por las aplicaciones.
- Evitar contabilizaciones intermedias y situaciones "en tránsito".
- Hacer uniformes los procedimientos, circuitos y procesos.
- Contabilizaciones automáticas y en cuentas restringidas.
- Tratamiento operativo distinto entre:
  - Operaciones rápidas → pregunta/respuesta.
  - Operaciones pesadas → en diálogo.

- Antes de dar soluciones con estructuras de Bases de Datos evaluar la posibilidad de usar caminos alternativos sencillos como archivos planos o indexados y procesos secuenciales.
- Empleo de interacción a través de periféricos sólo cuando sea imprescindible.
- Considerar controles de prevención y detección de incidencias o fallas.

#### **Operación en Oficinas.**

- Se elimina información de salida en "papel", creándose listados como opción de salida para la utilización en oficinas, integrándose en el sistema general de distribución.
- No fundamentar la solución sólo en el sistema, considerar otros elementos de solución complementarios (como son los operativos).
- Se genera información histórica en medios informáticos accesibles en tiempo real, sin necesidad de generar informes de salida.
- Evitar trabajos voluminosos y repetitivos, que no aporten valor a la gestión de productos y operaciones, reduciendo el "back office".
- Desactivar la "especialización" por funciones / estructuras, tratando de universalizar tratamientos en cualquier puesto.
- Eliminar las tareas de control sobre procesos y operaciones en los que no interviene la tarea.
- La documentación dirigida a clientes se debe integrar al sistema de correspondencia.
- Las ejecuciones de transacciones tienen un resultado "final", aceptada o rechazada.
- Eliminar la manipulación de papeles y listados para la ejecución de operaciones.
- Evitar tareas de control para operaciones "no finalizadas".
- Las incidencias "corregidas" son por correcciones de raíz.
- Evitar hacer varias acciones para una sola operación.
- Eliminar apuntes contables manuales.

### 4.1.3 Estándares CDR

Las líneas generales a considerar para el diseño dentro del CDR se dividen en dos ámbitos importantes:

- CDR, con relaciones hacia los grupos de fábrica de software y pruebas de fábrica.
- CCR, con relaciones hacia los grupos de infraestructura, calidad y producción.

En el ámbito CDR la definición de estándares se centra en los siguientes aspectos:

- Lineamientos de programación on-line (en línea) y batch (por lotes).
- Nomenclatura de componentes.
- Paquetización<sup>20</sup> y paralelismo.
- Definición de programas.
- Control de versiones.

En el ámbito de relación CDR / CCR, se abordan los estándares en los siguientes aspectos:

- Manejo de utilerías y versiones de programas producto.
- Lineamientos gestión de espacios.
- Lineamientos de la base de datos.
- Lineamientos on-line.

---

<sup>20</sup> Paquetización: término coloquial usado principalmente para definir que los programas se agrupan en paquetes de acuerdo a sus características comunes o funcionalidad.

#### 4.1.3.1 Definición de programas

Creación de un flujo de información eficiente para el trabajo con la fábrica de software, apoyado principalmente en un documento que permita la definición de lo que debe hacer el programa a realizar. La estructuración de dicho documento debe contener:

- **Objetivo:** Código y nombre del programa con una explicación breve del mismo.
- **Contexto:** Tanto para programación on-line como batch se debe aportar un plano de situación adecuado para facilitar la comprensión y agilizar la construcción por parte de los recursos receptores del mismo; analistas y programadores de la fábrica de software.

Contendrá los siguientes elementos:

- Objetos relacionados de interacción, ya sean bases de datos, on-line, arquitectura o archivos de entrada/salida.
  - Modelo de programa similar, cuando aplique.
  - Interfases de relación internas y/o externas.
  - Inicio y dependencias de ejecución.
  - Tipo y subtipo de programa.
  - Área de comunicación.
  - Método de llamada.
- 
- **Lógica de programación:** Debe contener información precisa, concisa y clara, con tal nivel de detalle, que el responsable de codificar no requiera ninguna otra información.

Considerar:

- Manejo de objetos y control de errores.
- Funcionalidad contenida.
- Cálculos.

- Control de interfases internas y externas, y hacia usuarios.
  - Evitar pseudo-código salvo necesidades expresas de:
    - a. Requerimiento de diseño.
    - b. Complejidad de proceso o cálculo.
    - c. Rendimiento aplicativo.
  - Secuencia lógica de proceso de aplicación.
- 
- Las solicitudes formales de trabajos a la fábrica de software y pruebas CDR se manejarán bajo el formato de algún documento que permita llevar el **control de componentes**. La función de este documento será la agrupación de componentes (paquetización) enviados a la fábrica de software y a pruebas CDR en unidades lógicas que permitan ser construidas y probadas independientemente, a fin de lograr mayor paralelismo y rapidez en la construcción de soluciones, así como una clara expresión de flujos y secuencias de la solución planteada. Además será utilizado para el seguimiento y detección de desviaciones durante el desarrollo de la aplicación.

### 4.1.3.2 Paquetización y paralelismo

Con la agrupación en paquetes de componentes, también se define el flujo de objetos y a su vez el flujo general del desarrollo del proyecto. Además se consigue lograr paralelismo en la construcción y en las pruebas, lo que se traduce en reducción de los tiempos totales de desarrollo.

El efecto mencionado se muestra a continuación (ver figura 4.2):

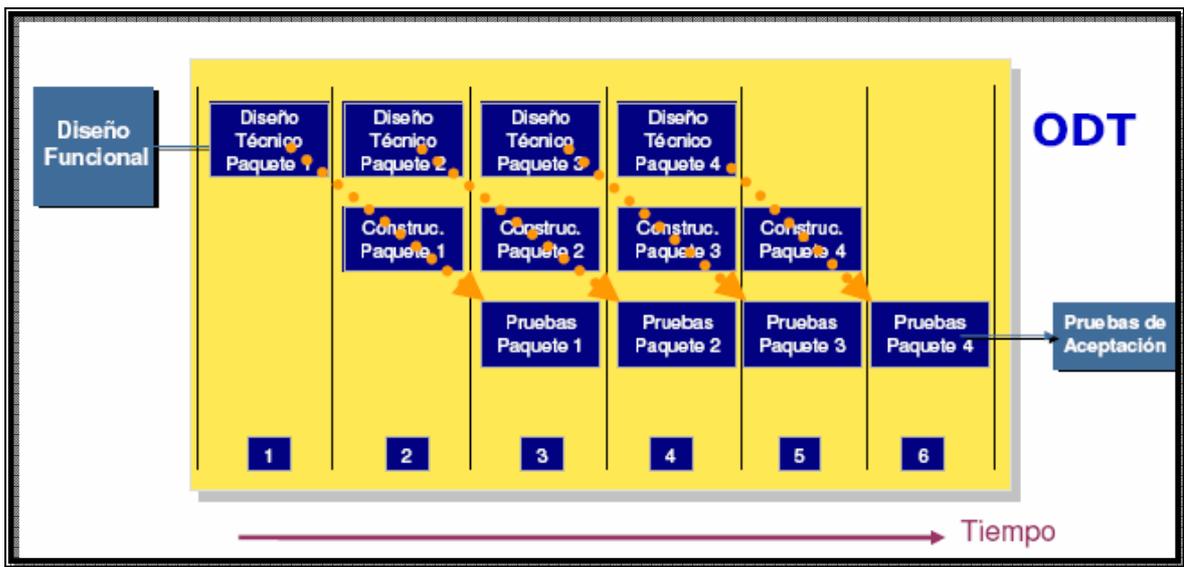


Figura 4.2 Paquetización y paralelismo

Otro punto importante a realizar dentro de esta etapa, es la identificación de versiones, esto consiste en que a partir de las solicitudes que ingresen al CDR y de acuerdo a la prioridad dada por el usuario, la naturaleza y la complejidad de la modificación, las solicitudes se agrupen y en su conjunto formen una versión.

#### A. Diseño Técnico.

**Responsable:** CDR Área de Diseño Técnico.

Partiendo del diseño funcional, el área de diseño técnico se encargará de realizar un análisis de viabilidad y detallar el diseño técnico en las especificaciones correspondientes. Considerando como parte del mismo, el diseño de las pruebas técnicas, los procesos de instalación y la estrategia de implantación de la solución.

Es importante que se entregue al mismo tiempo, tanto el diseño técnico como la matriz de pruebas a las áreas de fábrica de software y pruebas CDR.

#### **B. Construcción.**

**Responsable:** CDR Fábrica de Software.

En esta etapa se realiza el desarrollo y pruebas de unidades lógicas.

#### **C. Pruebas.**

**Responsable:** CDR Pruebas CDR.

En esta etapa se realiza las pruebas en el ambiente de calidad y se prueba tanto las unidades lógicas, como el sistema en su totalidad.

#### **D. Implantación (Piloto y Seguimiento Post-Liberación).**

**Responsable:** CDR Diseño Técnico.

Liberación de la aplicación y solución de las incidencias dentro del período de garantía.

#### **E. Fase de Liberación.**

**Responsable:** CDR, CCR y desarrollo locales.

Envío de la aplicación a los países que la requieren y efectuando su liberación a producción.

**Nota:** Es importante tanto en la etapa de implantación como en la de fase de liberación, contar con el compromiso y la colaboración de la institución en el país donde se este liberando un producto para que éste resulte exitoso.

#### **F. Soporte a Países.**

**Responsable:** CDR, CCR y desarrollo locales.

Corrección de incidencias dentro del período de garantía cuando se trata de una versión nueva y solución a problemas en producción.

### 4.1.3.3 Control de versiones

Con la agrupación en paquetes la identificación de control de versiones de aplicaciones, bajo el concepto de paquetización antes descrita, será identificada y controlada en base a la siguiente nomenclatura y contenidos (ver ejemplo figura 4.3):

- Toda versión aplicativa se reconocerá por la terna **VERSION.RELEASE.FIX** donde cada uno de ellos se expresará como dígitos numéricos secuenciales con orden de precedencia:
  - Primero VERSION
  - Segundo RELEASE. Cada vez que cambie el precedente VERSION, el RELEASE inicia con cero.
  - Tercero FIX. Cada vez que cambie el precedente RELEASE, el FIX inicia con cero.
  - Ejemplo: Mis Clientes Versión 1.0.0 evolucionaría según las secuencias
    - 1.0.1 (cambio de FIX).
    - 1.1.0 (cambio de RELEASE).
    - 1.1.1 (cambio de FIX).
    - 1.1.2 (cambio de FIX).
    - 1.2.0 (cambio de RELEASE).
    - 2.0.0 (cambio de VERSION).

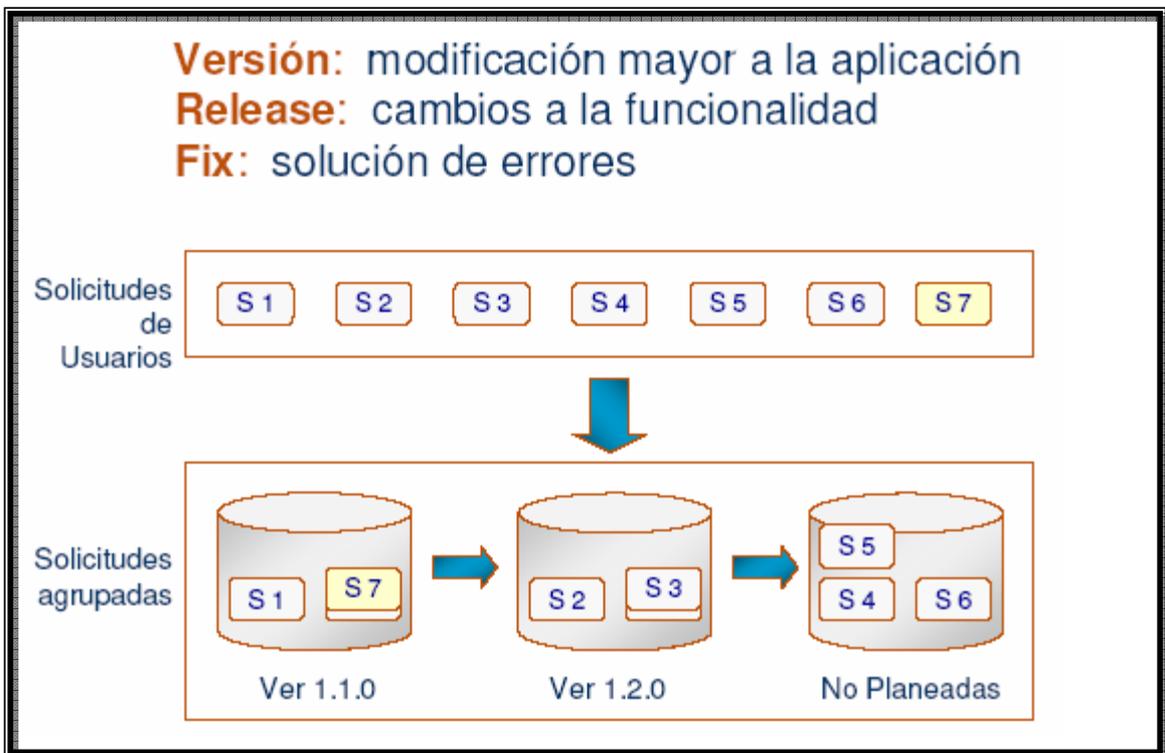


Figura 4.3 Ejemplo de control de versiones

#### 4.1.3.4 Control de versiones de componentes

La identificación de control de versiones de componentes, siguiendo los conceptos de paquetización y nomenclatura antes descrita, se recomienda hacerlo de la siguiente forma:

Se utilizará el último dígito significativo del nombre del programa para su identificación. Ej.: BG2C5030 (versión 0) pasa a ser BG2C5031 (versión 1) (ver figura 4.4).

El máximo número de versiones utilizadas será de 4 en secuencia circular, es decir, las versiones de programas iniciarán en versión 0 e irán incrementando la secuencia de versión hasta 3 para volver a utilizar la versión 0.

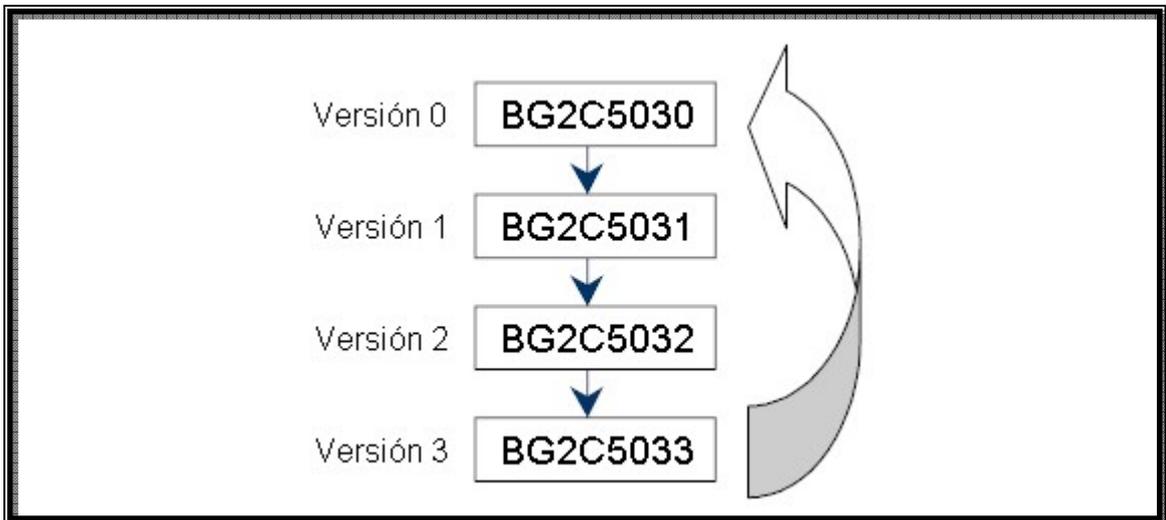


Figura 4.4 Ejemplo de control de versiones de componentes

Se utilizan inicialmente sólo cuatro versiones, 0 a 3, debido al impacto que este concepto tiene sobre el crecimiento en tamaño de las librerías de carga de todos los ambientes de desarrollo y producción. El concepto es aplicable a todo tipo de componentes (programas) que sean creados durante el proyecto.

### 4.1.3.5 Nomenclatura de componentes

Dentro del universo de programas existen varios tipos, como ya se había mencionado se encuentran los de línea (mejor conocidos como programas on-line), los de tipo batch (también conocidos como programas por lote), algunos más se conocen como rutinas y otros son empleados como programas que sólo almacenan variables de uso común también llamados (copys).

Debido a esto la necesidad de mantener un control sobre los programas se hace evidente, más aún cuando un sistema por lo general esta compuesto por muchos tipos de programas distintos interactuando entre si, por lo cual resultaría ideal que existiera una referencia que de acuerdo a las características del programa que se este desarrollando ayudará a determinar la forma correcta de nombrar cada componente de modo que con sólo ver el nombre se puede conocer el tipo de aplicación que es, el lenguaje usado en su construcción, la versión que corresponde, etc.

A manera de sugerencia se presenta la tabla 4.1 en la cual se puede observar una forma adecuada de nombrar cualquier componente de sistemas, la nomenclatura sería AATLDDDDV donde:

AA	Aplicación
T	Tipo de programa 1: Programa Línea sin acceso a BD* 2: Programa Línea con acceso a BD 3: Programa Batch sin acceso a BD 4: Programa Batch con acceso a BD
L	Lenguaje del programa A Assembler C Cobol II V Cobol VS J Java
DDD	Nemónico de uso del programa
V	Versión del programa
*	BD (Base de datos)

Tabla 4.1 Ejemplo de nomenclatura para componentes

#### 4.1.4 Estándares guía para el área de fábrica de software

El **objetivo** del área de fábrica de software es crear aplicaciones, para el CDR (Centro de Desarrollo Regional) a un menor costo, con un alto nivel de calidad y confiabilidad, desarrollando bajo un esquema estándar de trabajo y con una metodología que permita ofrecer los niveles de confiabilidad requeridos.

Las **actividades** que realiza son:

- Certificación de calidad revisando que los productos cumplan con las características determinadas por el CDR, para las pruebas individuales, que incluyen lógica, documentación, reutilización y funcionalidad.
- Verificación de diseño técnico, que éste sea claro, completo, sencillo, lógico y estructurado bajo los estándares del CDR.
- Construcción de soluciones mediante estándares de lógica, de codificación y documentación.

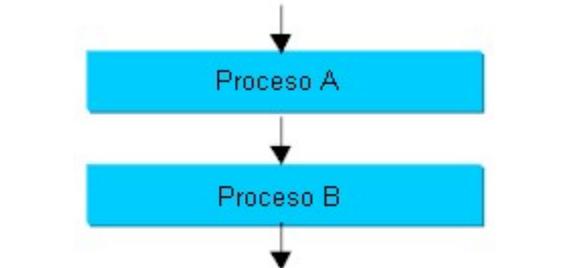
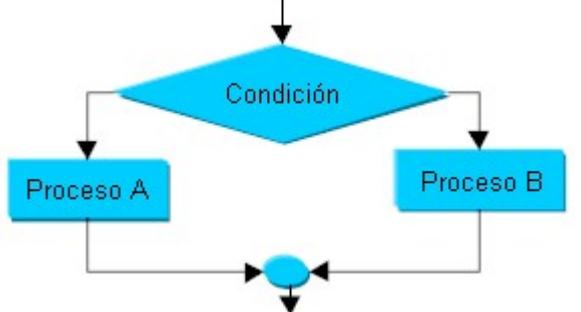
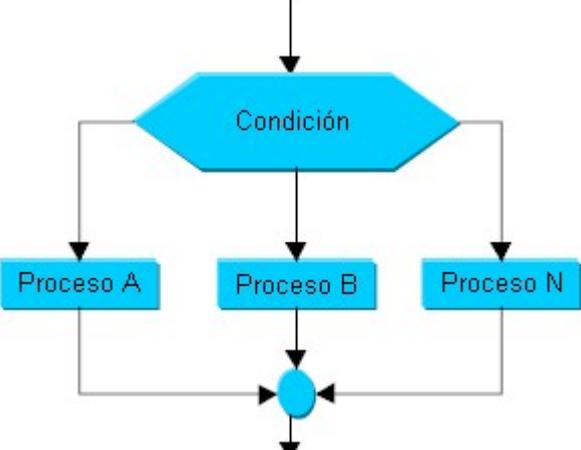
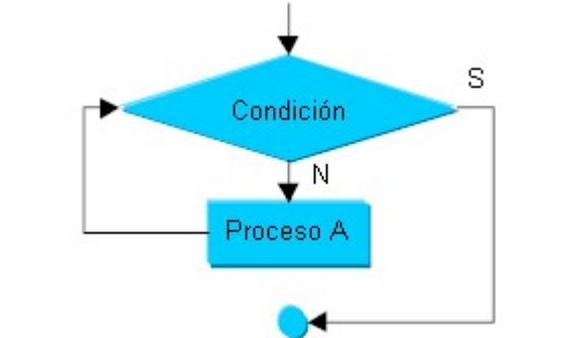
A partir del siguiente esquema:

El equipo de construcción desarrollará la creación de componentes bajo estándares de lógica, basándose en 6 estructuras lógicas de programación, nomenclatura estándar para variables internas y secciones del programa así como el uso de instrucciones de más bajo nivel acorde a las especificaciones realizadas por el equipo de diseño, con la finalidad de asegurar la legibilidad de los productos, asegurando así la independencia del producto al programador.

Así mismo el equipo realizará las pruebas unitarias al producto desarrollado de acuerdo a los requerimientos mínimos de la expectativa.

### 4.1.4.1 Estructuras lógicas

Las estructuras lógicas mas comunes se muestran a continuación en la tabla 4.2:

<p><b>1.- Secuencia</b></p> <p>Ejecución de 2 o más procesos en forma consecutiva          Un proceso puede verse como instrucción o conjunto de instrucciones ligadas dentro de un componente.</p>	
<p><b>2.- Decisión sencilla (IF)</b></p> <p>Ejecución de uno de dos procesos dependiendo de una condición          Uno de los procesos puede estar vacío.</p>	
<p><b>3.- Decisión múltiple (CASE)</b></p> <p>Ejecución de más de 2 procesos dependiendo de una condición          Uno de los procesos puede estar vacío.</p>	
<p><b>4.- Mientras (WHILE_DO)</b></p> <p>Ejecución de un proceso varias veces hasta que no se cumpla una condición, si esta se cumple de inicio no se ejecuta ni una sola vez.</p>	

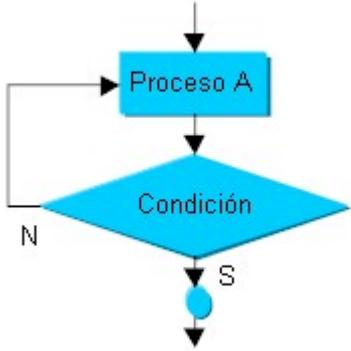
<p><b>5.- Repite (REPEAT-UNTIL)</b></p> <p>Ejecución de manera repetitiva mientras no se cumple una condición que se evalúa después de la ejecución de un proceso.</p>	
<p><b>6.- Salida anormal (Aborte)</b></p> <p>Finaliza el programa cuando se detecta una condición anormal a la operación de un componente.</p>	
	

Tabla 4.2 Estructuras lógicas

#### 4.1.4.2 Estándares de acceso a base de datos

##### *Programación.*

- Se evitará codificar muchas veces la misma sentencia, codificarla una sola vez e incluirla en las rutinas del programa, y llamarla las veces que sea necesario pasándole los diferentes valores en las variables de entorno.
- Se evitarán operaciones aritméticas dentro de las sentencias de consulta, será mejor primero resolver la operación aritmética usando lógica de programación y después ejecutar la consulta pasando el resultado de la operación aritmética en las variables de entorno.
- Se evitará usar cláusulas abiertas para buscar información, las sentencias deberán ser muy explícitas sin dejar condiciones de más, sólo las necesarias.
- En las condiciones de acceso a la base de datos se codificarán primero aquellas que vayan a filtrar más datos y posteriormente aquellas que filtren menos.

### 4.1.5 Estándares guía para el área de pruebas CDR

En todo desarrollo de sistemas existe una fase importante y necesaria que permite determinar si los resultados obtenidos son los que se esperaban o no, esta es la fase de pruebas y siendo tan relevante no se debe omitir bajo ninguna circunstancia, lo cierto será que si no existe una matriz o alguna línea a seguir que permita realizar pruebas apegadas a la realidad y que puedan ser confiables, no se podrá estar seguro si lo que se obtuvo era lo que se esperaba en verdad o no (ver figura 4.5).

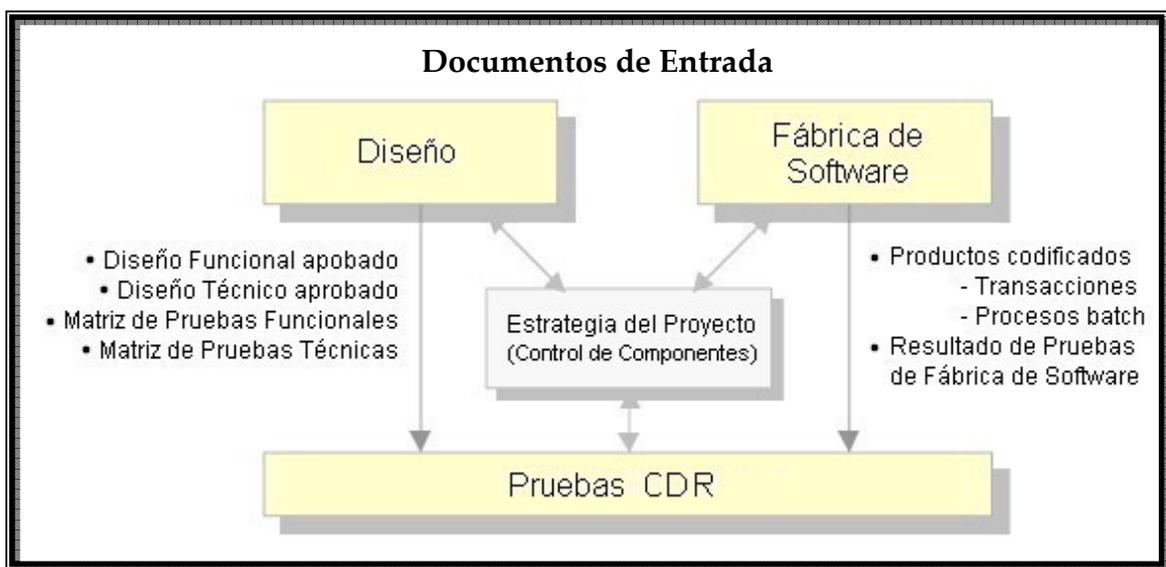


Figura 4.5 Estándares guía para el área de pruebas CDR

No existen propiamente herramientas 100% efectivas que puedan ayudar a realizar pruebas, ya que aunque todos los sistemas mantienen cierta similitud en su diseño y funcionalidad, la mayor parte de las veces se presentarán inconvenientes específicos por cada caso de error y la forma de corroborar la existencia de alguna falla tendrá que verificarse desde diferentes ángulos que por necesidad serán exclusivos por cada suceso.

Por tal motivo las siguientes recomendaciones o guías determinan brevemente los puntos clave que deberán estar presentes a la hora de llegar a esta fase que es realmente la etapa donde se podrá “garantizar” un buen resultado y localizar, en caso de que se presente, algún error en el momento justo donde se produce y corregirlo.

#### 4.1.5.1 Definición de la estrategia del proyecto (diseño, construcción, pruebas)

Resulta importante no perder de vista los siguientes puntos:

- Elaborar plan del proyecto:
  - Establecer fechas de diseño.
  - Establecer fechas de construcción.
  - Establecer fechas de pruebas.
  - Estimar tiempos de construcción de entornos para cada etapa de la estrategia.
- Determinar indicadores de medición y aprobación.
- Definir secuencia de diseño y construcción:
  - Priorizar componentes.
  - Priorizar ensambles.
- Identificar requerimientos de entornos.

#### 4.1.5.2 Revisar y complementar matrices de pruebas

- **Consideraciones:**
  - El 80% de las condiciones y casos de prueba serán para verificar el 20% de los componentes, ensambles, funciones o procesos críticos de la aplicación, identificados por el área de diseño.
  - Cuando se vaya a probar una nueva versión o cambios por mantenimiento de una aplicación existente, se deberán ejecutar pruebas de regresión (prueba cero) para verificar que funciones y procesos ajenos a los cambios no fueron afectados. Esta tarea requiere la reutilización de matrices de pruebas.
- **Validar y complementar la matriz de pruebas funcionales:**
  - La matriz de pruebas funcionales se valida contra el diseño funcional y la estrategia del proyecto (diseño, construcción y pruebas).
  - Para cada función y proceso de la aplicación deberá existir por lo menos una condición y caso de prueba.

- La matriz de pruebas debe considerar como mínimo las siguientes premisas básicas:

**Línea.**

- Control de accesos (seguridad) a nivel transacción, función y campos.
- Cumplimiento de estándares de pantallas.
- Funcionalidad de transacciones.
- Con datos de entrada erróneos.

**Batch.**

- Con archivos de entrada correctos, erróneos, vacíos e inexistentes.
- Interfases con aplicaciones internas y externas.
- Procesos de respaldo, reinicio y recuperación.
- Procesos de implantación de la aplicación.
- Cumplimiento de estándares de reportes.
- Funcionalidad de procesos.

• **Validar y complementar la matriz de pruebas técnicas:**

- La matriz de pruebas técnicas se valida contra el diseño técnico y la estrategia de diseño, construcción y pruebas.
- Se deben considerar condiciones y casos de prueba para verificar:
  - Cumplimiento de estándares de diseño y construcción.
  - Lógica y estructura interna de programas (segmentos, ramas, condiciones, ciclos, sentencias de acceso a bases de datos, instrucciones y manejo de códigos de retorno).

- También se debe considerar la evaluación de los siguientes aspectos de rendimiento:

**Línea.**

- Consumo de recursos a bases de datos y acceso a pantallas.
- Cantidad de datos transferidos por unidad de tiempo.
- Tiempo de respuesta.

**Batch.**

- Cantidad de datos transferidos por unidad de tiempo.
- Espacio en disco utilizado (temporal o permanente).
- Consumo de recursos de acceso a bases de datos.
- Consumo de memoria y procesador.
- Tiempo de ejecución.

### 4.1.5.3 Preparar entorno y datos de pruebas

**Consideraciones:**

- Preparar el entorno:
  - Solicitar a las áreas correspondientes la instalación o configuración del hardware, software, herramientas, aplicaciones o arreglos físicos necesarios.
  - Consultar los requerimientos para el desarrollo de cada una de las pruebas, documentados en la estrategia de diseño y construcción.
  - Solicitar al área correspondiente los usuarios y accesos que serán requeridos.
  - Aprovechar el conocimiento y experiencia adquirida durante las pruebas de fábrica de software.
  - Verificar el entorno de pruebas:
    - Deberá reproducir en la medida de lo posible las condiciones del entorno de producción.
    - Las diferencias significativas entre el entorno de pruebas y el de producción deberán documentarse, así como los riesgos que ello implica.
- Preparar los datos para ejecutar las pruebas.
- Solicitar la descarga de datos del entorno de producción.
- Solicitar la carga de datos en el entorno preparado para probar.
- Verificar y validar los datos:
  - Deberán cubrir todas y cada una de las condiciones y casos de prueba especificados en la matriz para tal efecto.
  - Por seguridad, los datos obtenidos del entorno de producción deberán enmascarse.

#### 4.1.5.4 Ejecución de pruebas

##### Consideraciones:

- Cada vez que un defecto sea corregido, se tendrán que ejecutar nuevamente pruebas de regresión para asegurar que el defecto quedó corregido y que no se afectaron otras funciones o procesos.
  - Regresión parcial: probar sólo las condiciones y casos de prueba afectados.
  - Regresión total: volver a probar todas las condiciones y casos.
- Ejecutar pruebas de volumen con las transacciones y procesos batch que procesarán cantidades grandes de información.
- Analizar y comparar los resultados obtenidos contra los resultados esperados y documentar las discrepancias.
- Ejecutar las pruebas con cada una de las condiciones y casos documentados en las matrices correspondientes.
- Apoyarse en el área de calidad del CCR para ejecutar e interpretar el resultado de las pruebas técnicas.
- Reportar y documentar todos los problemas detectados durante la ejecución de las pruebas.
- Simular o provocar fallas externas y evaluar el comportamiento presentado.
- Realizar pruebas con más de un usuario participando simultáneamente.
- Asegurar que el entorno y los datos de prueba estén preparados.

#### 4.1.5.5 Administrar el proceso de pruebas

##### Consideraciones:

- Evaluar el cumplimiento de los criterios de inicio y terminación.
- Asegurar el logro de los objetivos y alcance de las pruebas.

- Criterios para iniciar la preparación de las pruebas:
  - Estrategia de diseño, construcción y pruebas aprobadas.
  - Matrices de pruebas elaboradas por el área de diseño.
  - Diseño funcional y técnico aprobados.
- Criterios para terminar la preparación de las pruebas:
  - Matrices de pruebas validadas y complementadas.
  - Entorno y datos de prueba preparados.
- Criterios para iniciar la ejecución de las pruebas:
  - Pruebas de fábrica de software verificadas.
  - Entorno y datos de prueba verificados.
- Criterios para terminar la ejecución de las pruebas:
  - Resultado global de las pruebas documentadas (ver figura 4.6).
  - Todas las condiciones y casos de prueba ejecutados.
- Asegurar la participación en tiempo y forma de áreas y personas involucradas.
- Manejar los problemas y conflictos relacionados con el entorno de pruebas.
- Actualizar el 'informe integral de calidad'.

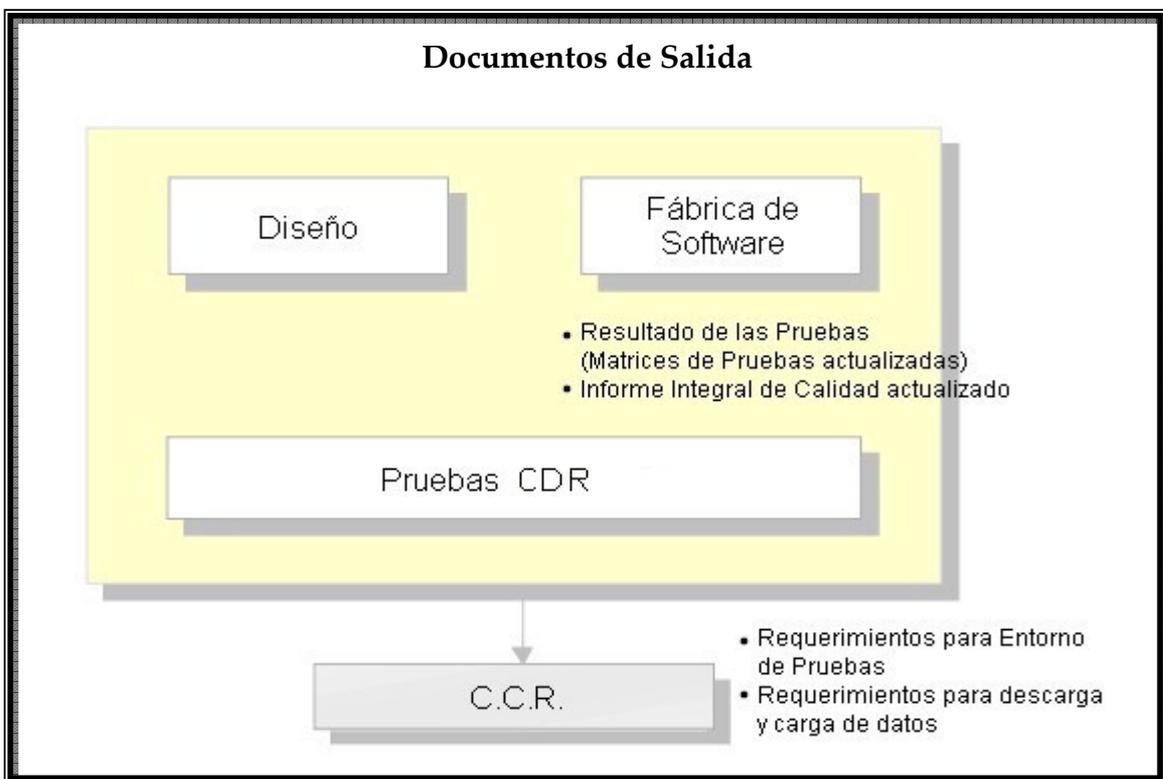


Figura 4.6 Documentación de salida en el área de pruebas CDR

#### 4.1.5.6 Construcción de entornos

##### Consideraciones:

- El entorno de pruebas deberá ser lo más parecido a producción, en algunos casos si es posible será conveniente crear un ambiente llamado “copia de producción”, que como su nombre indica, tendrá que ser una copia de lo existente en producción de modo que se tenga un grado de certeza mucho más real en los resultados que obtengamos.
- Toda actualización de tablas, archivos y bases de datos que son comunes entre aplicaciones deberá ser aprobada por las áreas que estén realizando pruebas.
- En las transacciones de prueba se deberán usar claves de usuarios que correspondan a niveles reales de facultades evitando así hacer uso de súper usuarios.
- Durante el período de pruebas, el entorno deberá ser periódicamente actualizado con los procesos diarios, quincenales o mensuales que sean necesarios.
- Las limitaciones y diferencias respecto al entorno de producción deberán quedar claramente especificadas en el plan de pruebas.
- Los datos y cuentas que se utilicen en las pruebas deberán ser de uso exclusivo de los usuarios que las estén realizando.
- Los datos y cuentas de prueba deberán ser enmascarados respecto a su origen en producción.

#### 4.1.5.7 Pruebas CDR (Centro Desarrollo Regional)

##### Consideraciones.

- **De pruebas en transacciones en línea:**
  - Todas las pantallas de la aplicación deberán estar apegadas a estándares de diseño en cuanto a distribución, uso de colores, longitud y ubicación de los mensajes de error, utilización de las teclas función y utilización de campos llave.
  - Se deberá ejecutar al menos una prueba en la que participen simultáneamente tantos usuarios y perfiles como sea posible.

- Cada vez que uno o más problemas sean identificados y corregidos, se deberán ejecutar pruebas de regresión parciales o totales identificando si es necesario volver a probar sólo las condiciones y casos de prueba afectados.
- Cuando una transacción encadene o enlace con otra se deberán probar todas las posibles combinaciones de encadenamiento.
- En caso de contar con un manual de usuario se deberá validar los procedimientos y procesos ahí especificados.
- **De pruebas de procesos batch:**
  - Se deberá correr al menos un ciclo encadenado de cada tipo (ejemplo dos diarios para ejecutar un mensual, dos mensuales para correr un anual etc.)
  - Todos los reportes de la aplicación deberán estar apegados a estándares de diseño en cuanto a distribución y encabezados.
  - Se deberá probar la ejecución de procesos en fechas de fin de semana, días inhábiles, fechas pasadas y futuras.
  - Se deberán ejecutar procesos que provoquen saltos de hoja en todos los reportes de la aplicación.
  - Se deberán probar todos los puntos de reinicio y recuperación de una cadena.
  - Se deberán probar procesos con el volumen máximo esperado.
- **De procesos de cálculo:**
  - Se deberá verificar que los cálculos excesivos sean ejecutados en procesos batch y justificar plenamente los que se realicen en línea.
- **De niveles de seguridad:**
  - La prueba de niveles de seguridad deberá iniciar utilizando las claves de usuarios de nivel inferior.
- **De pruebas de rendimiento:**
  - Se deberá asegurar que todas las transacciones de la aplicación no rebasan los estándares establecidos en cuanto a:
    - Espacio en disco utilizado y consumo de recursos de bases de datos.
    - Cantidad de datos transferidos por unidad de tiempo.
    - Creación y vigencia de archivos de respaldo.
    - Consumo de memoria.
    - Tiempo de respuesta.

#### 4.1.6 Relación del Centro Corporativo Regional con el CDR

El CCR es el Centro Corporativo Regional (ver figura 4.7) y como tal, es la sección donde se concentran todas las áreas de apoyo técnico para desarrollo y diseño tales como:

- Procesamiento y Tecnología.
- Recursos Humanos.
- Servicio a Clientes.
- Finanzas.

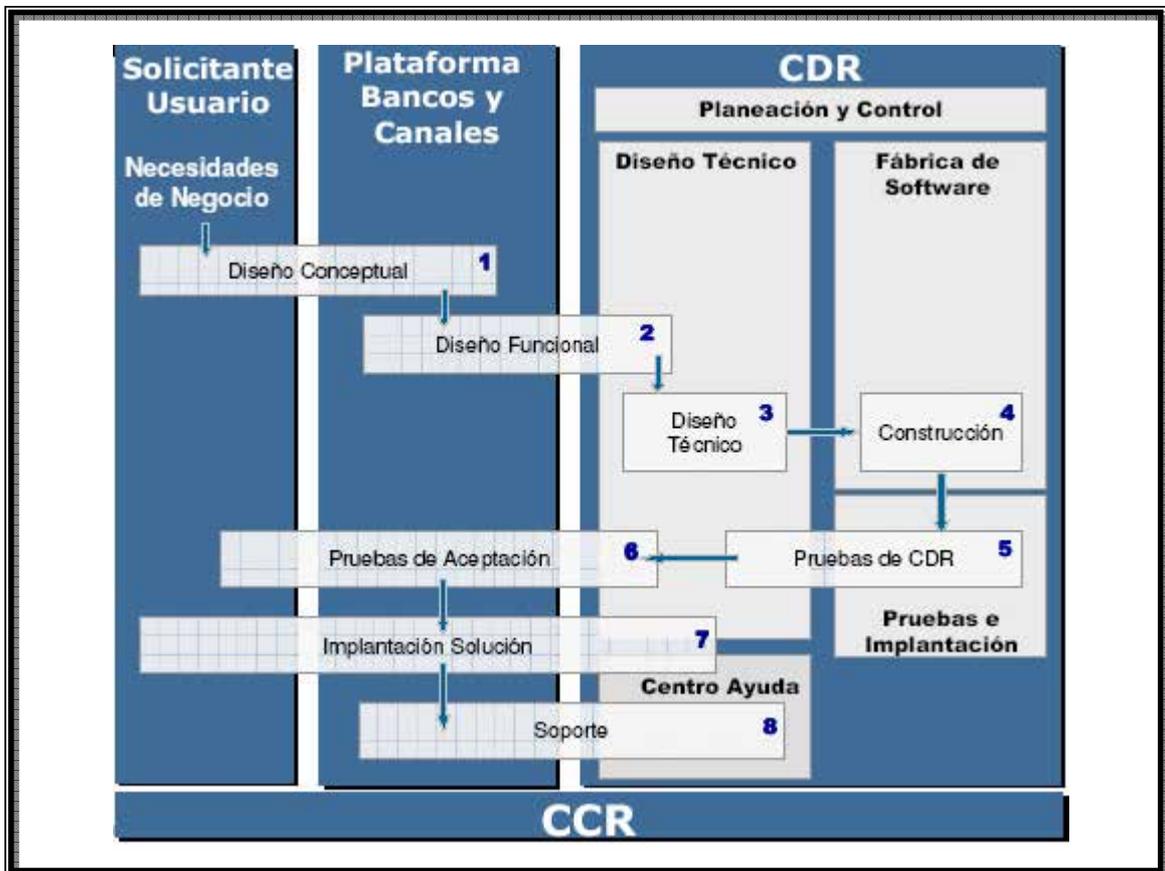


Figura 4.7 Relación del Centro Corporativo Regional con el CDR

Procesamiento y Tecnología esta formado por cuatro áreas importantes:

- Seguridad de Recursos Informáticos.
- Gestión de la Infraestructura.
- Procesamiento de Datos.
- Gestión de Calidad.

Las actividades del CCR son:

- La creación de infraestructura de acuerdo a las necesidades del Centro de Desarrollo Regional (ver figura 4.8).

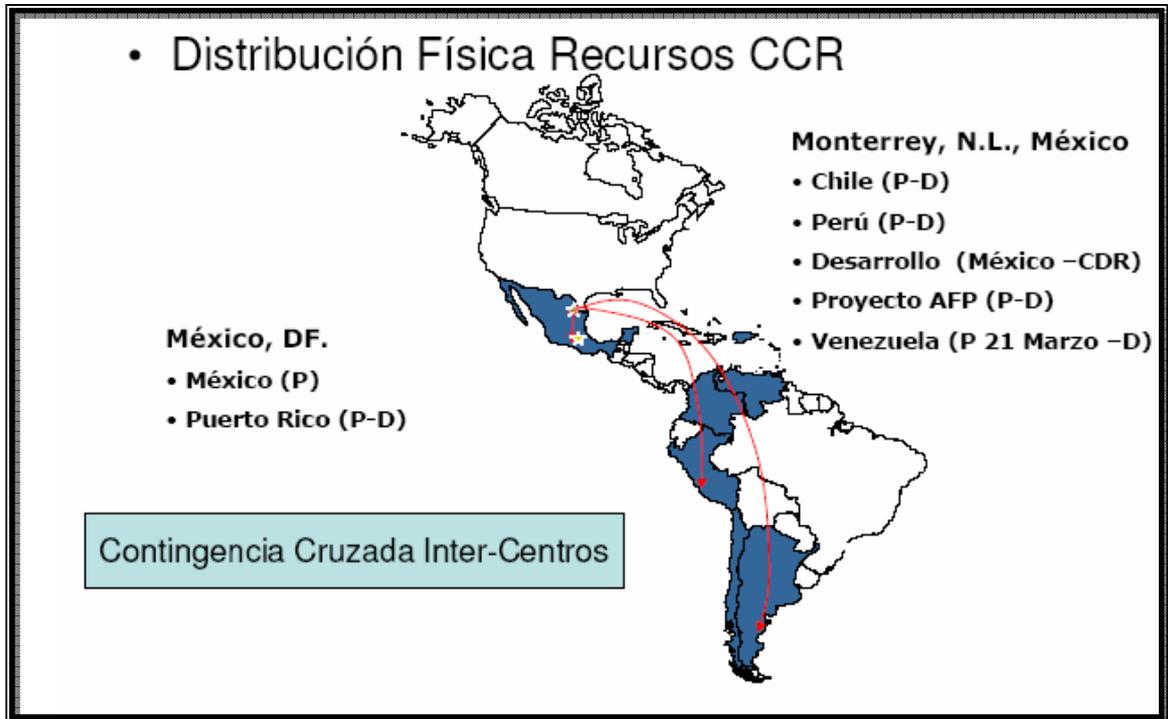


Figura 4.8 Distribución física de los recursos del CCR

En la figura 4.9 se muestra un ejemplo de la infraestructura antes mencionada.

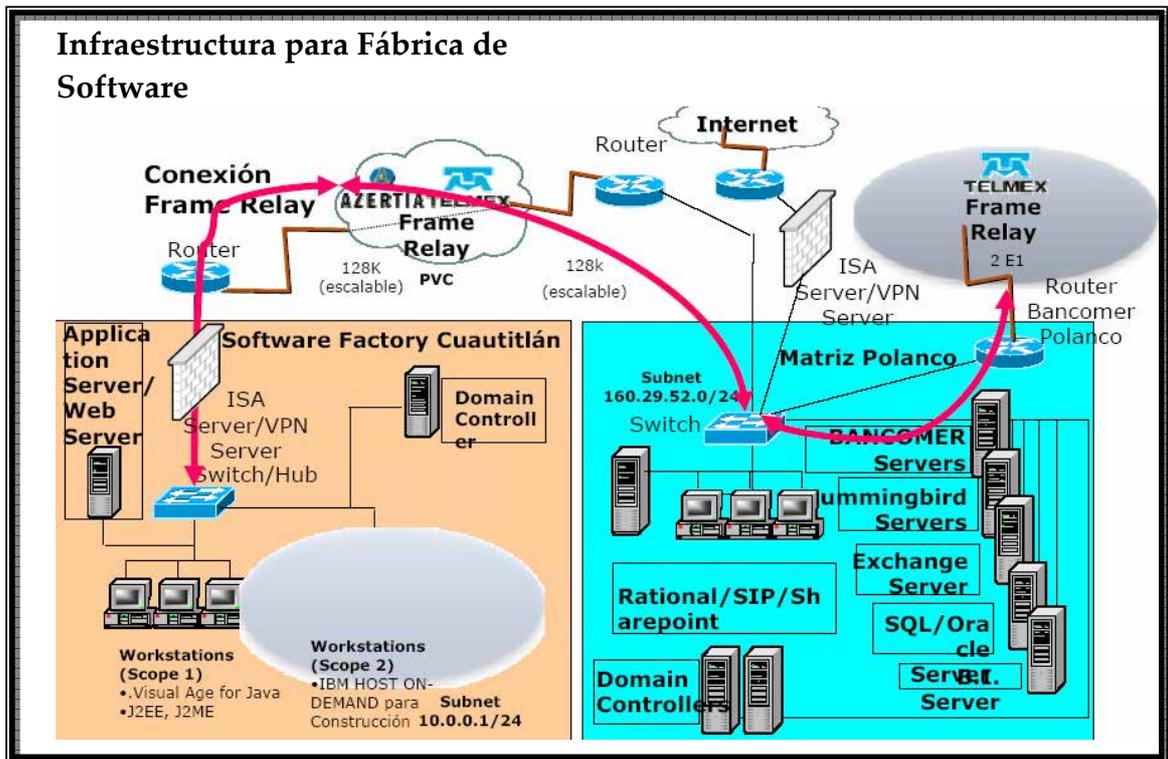


Figura 4.9 Ejemplo de la infraestructura para Fábrica de Software

- La creación de los entornos: Desarrollo-Test-Calidad (ver figura 4.10).

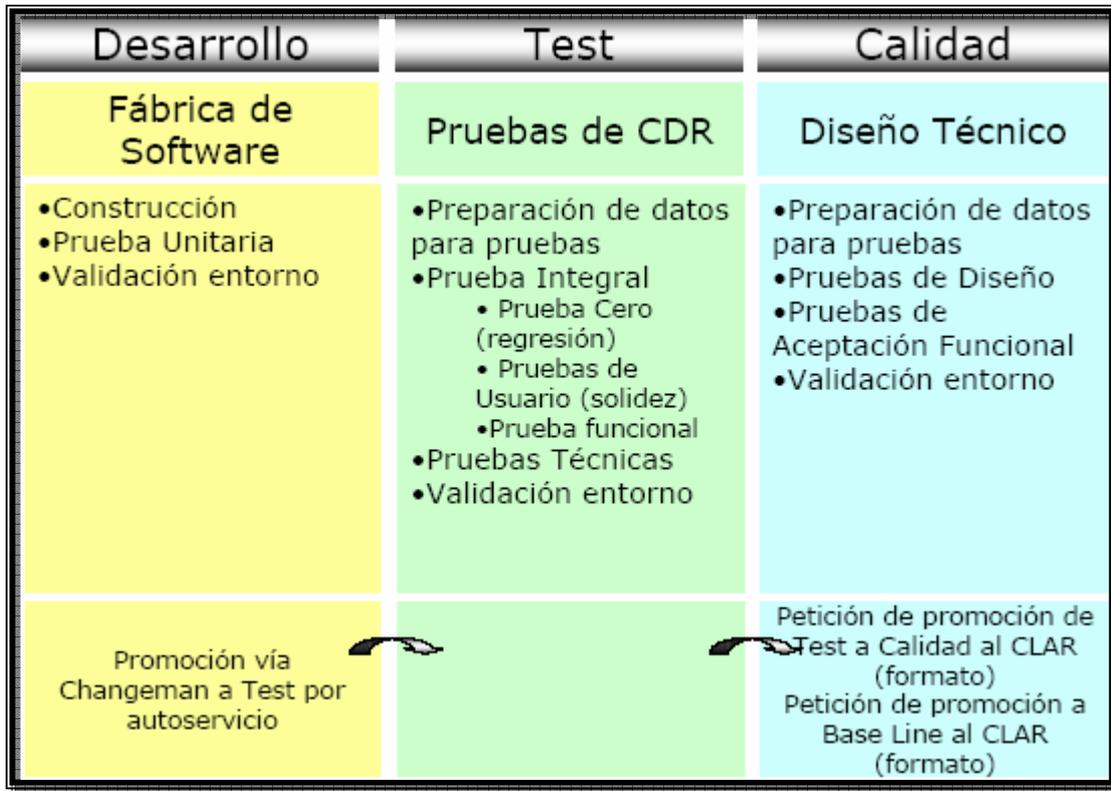


Figura 4.10 Ambiente Desarrollo – Test – Calidad

- Gestionar el proceso de transmisión de los aplicativos desarrollado en el CDR a los diferentes países donde se implementará el nuevo proyecto o sistema (ver figura 4.11).

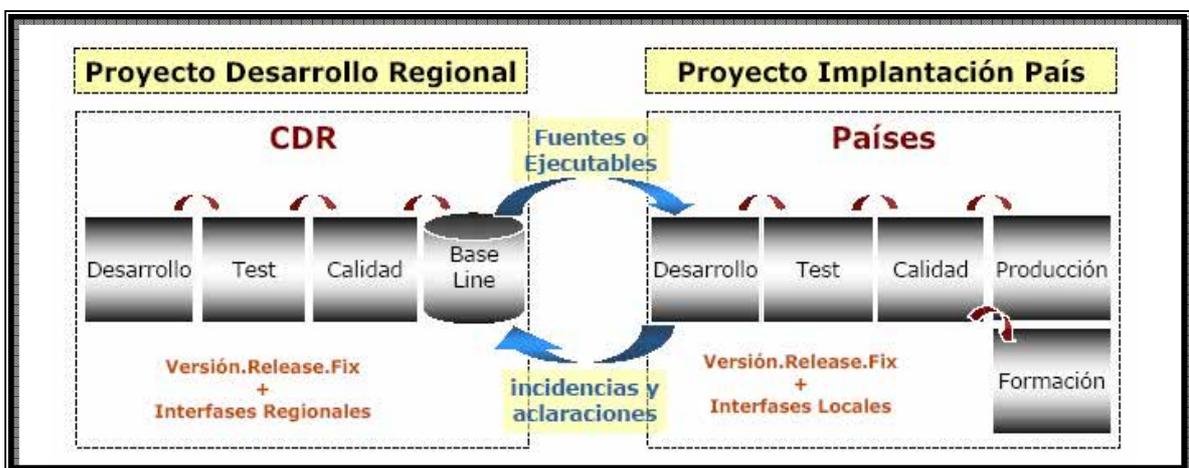


Figura 4.11 Proceso de transmisión de aplicativos por parte del CDR

- Crear los estándares del CCR para homologar la información de los diferentes países a concentrar (ver figura 4.12).

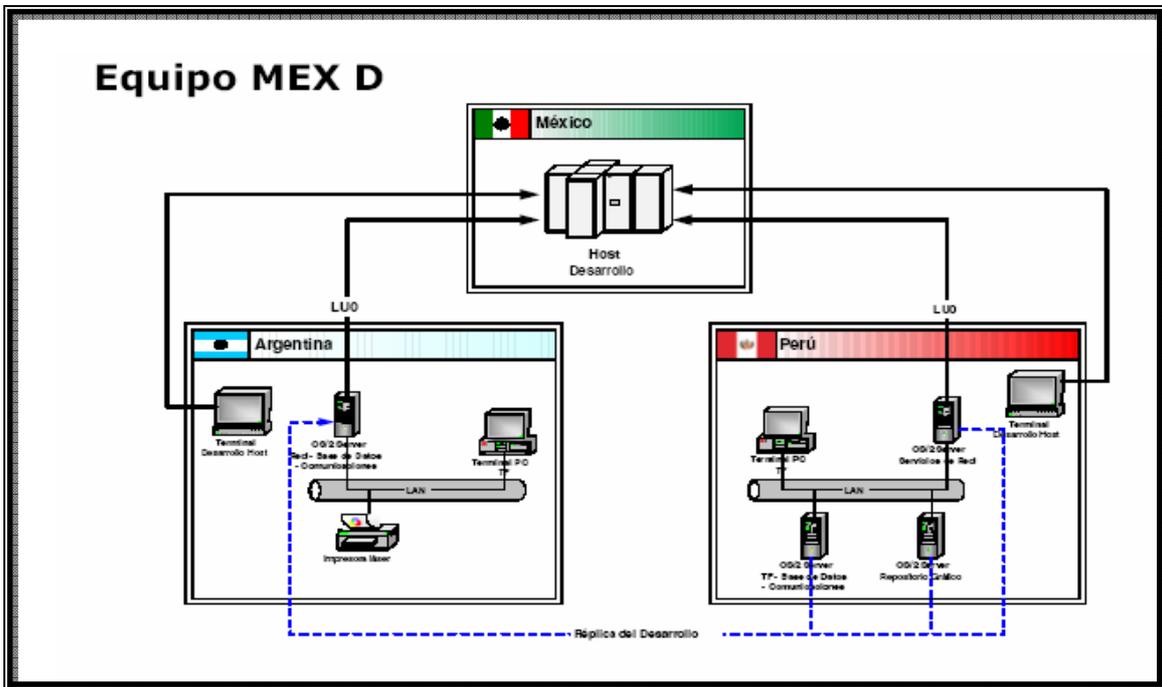


Figura 4.12 Estándares del CCR

Los aplicativos nuevos son conformados como se muestra a continuación en la figura 4.13.

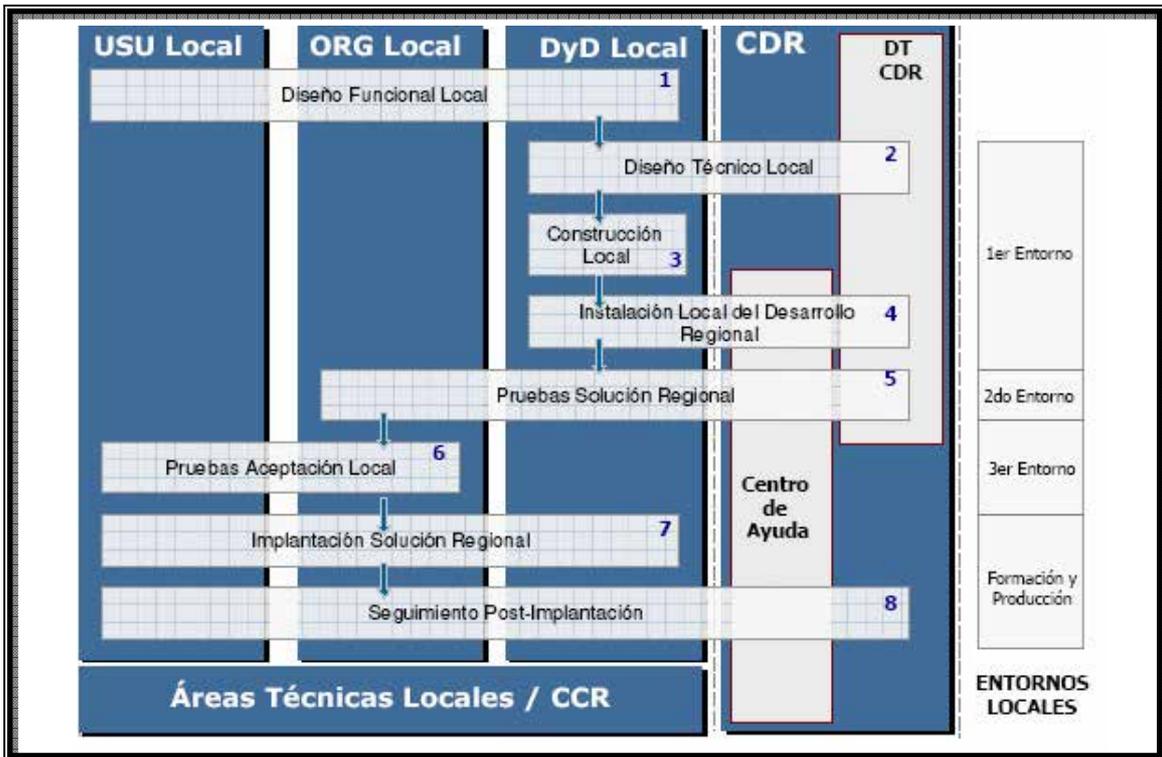


Figura 4.13 Modelo usado en la conformación de aplicativos nuevos

**La estrategia de calidad propuesta dentro de la institución financiera consiste en:**

- ❖ Implantar dentro del desarrollo y mantenimiento de aplicaciones un proceso de aseguramiento de la calidad basado en :
  - Uso adecuado de las prácticas sugeridas dentro del modelo de calidad CMMI.
  - Implantar prácticas de gestión de proyectos acordes en cada país.
  - Estándares funcionales como CDR y CCR.
  - Administración del conocimiento.
  - Comunicación efectiva.
  
- ❖ Incrementar el rendimiento de la organización tomando acción en:
  - Desarrollo de un plan de carrera.
  - Desarrollo de perfiles y puestos.
  - Clima organizacional.
  
- ❖ Diseñar y desarrollar una factoría de software en México que de soporte regional en Latinoamérica a través del desarrollo de un proceso corporativo para la administración del backlog<sup>21</sup> y de los requerimientos.
  
- ❖ Incrementar la capacidad de desarrollo reduciendo las actividades de mantenimiento.
  
- ❖ Reducir el tiempo para el lanzamiento de nuevos productos y servicios.
  
- ❖ Reducir los costos de operación.

Una estrategia acertada permitirá un despegue hacia un nivel superior de excelencia, hacia un mejor posicionamiento estratégico en la liga de los grandes grupos financieros globales y hacia la innovación sobre la base de un crecimiento rentable y el talento de los colaboradores.

---

<sup>21</sup> Backlog: valor de las órdenes pendientes de cumplimiento que tiene una compañía industrial, el aumento o la disminución de ese valor es clave para proyectar las cifras de ventas y ganancias.

# *Conclusiones*

---

---

Es claro que así como la complejidad de la construcción y mantenimiento del software ha aumentado en los últimos años, la complejidad de formar buenos ingenieros también. El ingeniero de software hoy se sitúa en el contexto de la empresa abierta, aquella que debe contar con sistemas de información que apoyen su negocio y con esto a sus clientes así como sus asociados. Estos son sistemas heterogéneos, altamente evolutivos y muy exigentes desde el punto de vista no funcional: deben escalar, ser altamente confiables, disponibles, seguros, ser capaces de manejar grandes volúmenes de datos y de transacciones, etc.

El ingeniero de software debe contar dentro de sus herramientas con los conocimientos y habilidades en los cuatro ejes de la ingeniería de software: procesos, arquitectura, metodología y tecnología. Debe ser capaz de encontrar un equilibrio entre ellos y poder aplicarlo en el momento de tomar una decisión dentro de una organización. El arquitecto de software que sólo maneje la tecnología de moda no podrá tener una visión completa de las consecuencias de sus decisiones. Por decir algo, el director de un proyecto que no maneje ninguna de las tecnologías de moda tampoco podrá juzgar adecuadamente las decisiones de los arquitectos.

Aplicando una adecuada estrategia de calidad en la ejecución de procesos, se podrá garantizar de forma confiable resultados más óptimos y apegados a las necesidades reales que exige el México actual, ya que tarde o temprano tendrá que ascender a niveles de calidad estándares en todo el mundo, para lo cual el uso de una metodología debe estar incluida en cada uno de los siguientes pasos para poder enfrentar los retos por venir.

Las universidades deben preocuparse por la preparación de sus estudiantes hacia un buen manejo del cambio. Inculcar la necesidad constante de actualización y mejoramiento individual. La educación formal debe preocuparse por impartir una base de conocimientos fundamentales y más aún por facilitar la generación de habilidades en los estudiantes que les permita aprender a moverse en las tecnologías cambiantes y en las empresas de hoy.

# Glosario

---

---

- ❖ **Calidad.**  
Propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor.
- ❖ **CCR.**  
Acrónimo de Centro Corporativo Regional.
- ❖ **CDR.**  
Acrónimo de Centro Desarrollo Regional.
- ❖ **CMMI.**  
Acrónimo de Capability Maturity Model Integration (Modelo integrado de madurez de capacidades).
- ❖ **Cronograma.**  
Calendario de trabajo.
- ❖ **Institucionalización.**  
Aplica a todo aquello que deba ser establecido como algo que debe realizarse al menos para garantizar la calidad o la correcta ejecución de cualquier procedimiento.
- ❖ **MPYMES.**  
Abreviación de micro, pequeñas y medianas empresas.
- ❖ **Paquetización.**  
Termino coloquial usado principalmente para definir que los programas se agrupan en paquetes de acuerdo a sus características comunes o de funcionalidad.
- ❖ **Paralelismo.**  
Término coloquial usado principalmente para definir que algo puede desarrollarse y/o construirse de forma paralela o dicho de otra forma, que ciertos procesos pueden generarse al mismo tiempo sin depender o afectarse mutuamente.
- ❖ **Priorización.**  
Termino coloquial usado en la acción de asignar prioridad a algo, en el caso de sistemas se refiere especificar en orden de importancia el desarrollo o ejecución de algún proceso.
- ❖ **Proceso.**  
Es el conjunto de fases sucesivas de una operación específica a realizarse.
- ❖ **Prosoft.**  
Es el programa diseñado por el gobierno mexicano para el desarrollo en la industria del software.
- ❖ **Qualitatem**  
Palabra en latín que significa atributo o propiedad que distingue a las personas, a bienes y servicios.
- ❖ **SEI.**  
Acrónimo de Software Engineering Institute (Institución en Ingeniería de Software).
- ❖ **Sistema.**  
Programa o conjunto de programas que efectúan la gestión de los procesos básicos de un sistema informático, y permite la normal ejecución del resto de las operaciones

# Bibliografía

- ✦ **HUMBERTO CANTÚ DELGADO**  
"Desarrollo de una cultura de calidad"  
Editorial: McGraw-Hill Interamericana Primera edición (1997)
- ✦ **MANUEL ABURTO JIMÉNEZ**  
"Administración por calidad"  
Editorial: CECSA-Compañía Editorial Continental (1992)
- ✦ **KAORU ISHIKAWA**  
"¿Qué es el control total de la calidad?: La modalidad japonesa"  
Editorial: Grupo Editorial Norma (1988)
- ✦ **JOSEPH M. JURAN**  
"Juran y la planificación para la calidad"  
Editorial: Ediciones Díaz de Santos (1990)
- ✦ **ARMAND V. FEINGENBAUM**  
"Control total de calidad"  
Editorial: CECSA-Compañía Editorial Continental (1998)
- ✦ **SARV SINGH SOIN**  
"Control calidad total: Claves, metodologías y administración para el éxito"  
Editorial: McGraw-Hill (1998)
- ✦ **HERBSLEB J. CARLETON, ROZUM J. SIEGEL AND ZUBROW D.**  
"Benefits of CMM based software process improvement: initial results"  
Software Engineering Institute (1994)
- ✦ **KOGUT B. AND TURCANU A.**  
"Global software development and the emergence of e-innovation"  
Carnegie Bosch Institute for Applied Studies in International Management (1999)
- ✦ **ROGERS, R.E.**  
"Implementation of total quality management: a comprehensive training program"  
Binghamton, New York, International Business Press (1996)

## Consulta en línea.

- ◆ <http://www.sei.cmu.edu/>  
Página oficial del "Software Engineering Institute" of Carnegie Mellon University.
- ◆ <http://www.qualitylink.com.mx/>  
Primer portal de calidad en México
- ◆ <http://www.software.net.mx>  
Sitio oficial de la Asociación Mexicana de la Industria de Tecnologías de Información.