

01167
8



UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA
DIVISION DE ESTUDIOS DE POSGRADO

PROCEDIMIENTO PARA LA ADMINISTRACION DE
PROYECTOS INFORMATICOS EN LA EMPRESA
ENLACES TECNOLOGICOS S. A. de C. V.

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERIA (PLANEACION)

P R E S E N T A :

CARLOS RINCON PARTIDA

DIRECTOR: DR. GABRIEL SANCHEZ GUERRERO



JULIO DEL 2001



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

En primer término quisiera agradecer a mis profesores y compañeros de la maestría por sus conocimientos, apoyo y amistad.

Al Dr. Gabriel Sánchez Guerrero por el tiempo que le dedicó a dirigir este trabajo y por sus consejos.

A mi familia.

Y finalmente a mi padre, a quien dedico este trabajo, por ser la inspiración para iniciar los estudios de posgrado.

Índice

1. Antecedentes	1
1.1 Descripción de la empresa Enlaces Tecnológicos S. A. De C. V.	1
1.2 Consideraciones sobre empresas similares	6
1.3 Problemática de mercado	15
1.4 Objetivo	18
1.5 Análisis de la empresa	20
1.6 Identificación de la necesidad de intervenir	24
2. Definición de los elementos del procedimiento para administrar proyectos en Enlaces Tecnológicos	25
2.1 Preanálisis (requerimientos)	25
2.2 Estipulación de criterios y lineamientos	27
2.3 Formato de propuesta	31
2.4 Análisis detallado	32
2.5 Diseño	39
2.6 Estimación	41
2.6-a Tiempo	41
2.6-b Recursos humanos	41
2.6-c Recursos tecnológicos	41
2.7 Definición de indicadores de avance	41
2.8 Definición de la forma de comunicación entre integrantes del equipo y con el cliente	42
2.9 Gestión de riesgos	42
2.10 Plan del proyecto	44
2.11 Construcción	46
2.12 Control	48
2.13 Pruebas de calidad	48
2.14 Entrega e instalación	51
2.15 Plan de seguimiento posventas	51
2.16 Dinámica para la especificación	52
2.17 Diagrama de flujo del procedimiento para la administración de proyectos informáticos en ET	53
3. Conclusiones	54
Apéndice I	56
Elementos de la administración de proyectos informáticos	56
Apéndice II	84
Motivación	84
Apéndice III	94
Análisis de opiniones	94
Bibliografía	101

1. Antecedentes

En este capítulo se hace una introducción de la empresa en la que se llevó a cabo este trabajo, una breve historia, sus unidades estratégicas de negocio (UEN), la forma en que está organizada, el entorno de empresas del ramo, los problemas que enfrenta y la necesidad de resolverlos y se describen, finalmente, los objetivos del presente trabajo.

1.1 Descripción de la empresa Enlaces Tecnológicos S. A. De C. V. (ET)

Enlaces Tecnológicos S.A. de C. V. tiene dos años y medio de haberse formado en Guadalajara, Jalisco y surgió a raíz de que un grupo de expertos en sistemas de software decidieron independizarse de las compañías en que trabajaban, a lo largo de este tiempo han participado en diversos proyectos para la banca, la industria y el comercio, con resultados favorables, lo que les ha llevado a ser reconocidos por sus clientes de tal forma que ahora la empresa cuenta con aproximadamente veinticinco empleados y participan en un gran número de proyectos, con una facturación anual de alrededor de seis millones de pesos.

Los servicios y productos que ofrece están organizados en las unidades estratégicas de negocio que se describen en la siguiente tabla:

Unidades Estratégicas de Negocio			
Servicios profesionales	Desarrollo de sistemas a la medida (soluciones integrales*)	Productos estándar	Escuela de cómputo y ciber-café
En este tipo de servicios la empresa no tiene bajo su control la administración del proyecto en el que participa, más bien vende el tiempo de los consultores y desarrolladores de sistemas para actividades concretas que requieran los administradores del proyecto.	En este caso la empresa tiene el control de la administración del proyecto y los objetivos del mismo están en función de los deseos del cliente. * Cuando se habla de soluciones integrales nos referiremos al caso en que la empresa ET contrata a otras compañías para algunas o todas las partes de la solución. En el caso contrario todo el desarrollo lo realiza ET.	Para los productos la empresa además de tener el control de la administración del proyecto también lo tiene de los objetivos, alcance y comercialización.	Los cursos que se ofrecen están en función de las condiciones de mercado, van desde básicos hasta los especializados en diferentes plataformas. Los servicios del ciber-café van desde la renta de equipos y conexión a Internet, hasta la búsqueda de información.

Tabla 1 Unidades Estratégicas de Negocio de la empresa

A continuación se describen brevemente los aspectos más relevantes de las UEN que se ilustraron anteriormente:

Servicios profesionales

La división de servicios profesionales fue creada para atender los requerimientos de empresas medianas y grandes, en materia de recursos humanos para proyectos que involucran sistemas de información. Disponen de grupos de técnicos especialistas en diversas tecnologías, - gente experimentada, capaz de una rápida integración a proyectos específicos, y con disposición al trabajo en equipo.

Adicionalmente, ofrecen servicios de consultoría, y capacitación de personal, a la medida

Tabla 2.- Descripción de la UEN Servicios Profesionales.

Desarrollo de sistemas a la medida (Soluciones integrales)
Levantamiento de información, análisis de procesos y requerimientos
Construcción y/o instalación de componentes e infraestructura de hardware y software
Pruebas, depuración e implantación de sistemas
Contratación de servicios de terceros, asesoría en la adquisición de herramientas y equipo
Paquetes de capacitación de usuarios y/o desarrolladores
Paquetes de mantenimiento y soporte técnico
Desarrollo de software para Microsoft Windows95/98 y 2000, Microsoft WindowsNT y Linux.
Desarrollo de software, soporte y asesoría en migración de sistemas BTOS/CTOS.
Diseño e implementación de soluciones de información sobre intranets y la Internet.
Servicios de instalación, configuración y mantenimiento de equipos de cómputo y redes.
Programadores de C++, Java y otros lenguajes de uso generalizado.
Desarrollo de aplicaciones de red y sistemas cliente/servidor sobre redes TCP/IP.
Desarrollo de software multiproceso, sistemas distribuidos y sistemas de misión crítica.
Desarrollo de drivers para dispositivos de hardware.
Experiencia en desarrollo de sistemas bancarios y financieros.
Experiencia en desarrollo de sistemas administrativos para la industria de la construcción.
Análisis y diseño de sistemas
Servicios de programación en C y Pascal, y scripting en JCL.
Desarrollo de software de conectividad con sistemas Microsoft y Unix.
Instalación, configuración y mantenimiento de estaciones y clusters.
Consultoría en proyectos de migración de sistemas y bases de datos.
Experiencia en desarrollo de sistemas bancarios y financieros.
Experiencia en customización de software (FSA Finesse, SignBank).
Soluciones bancarias en Edify.

Tabla 3.- Descripción de la UEN Desarrollo de servicios a la medida.

Productos estándar
Está unidad estratégica de negocio no ha contado con el desarrollo deseado por la administración de la empresa pero se tiene contemplado darle un impulso importante en el corto plazo. Algunos productos que se han desarrollado y que se pueden comercializar son:
Sistema de listas de raya para empresas constructoras.
Intranet para empresas.
Sistema de administración integral para restaurantes.

Tabla 4.- Descripción de la UEN Productos estándar.

Escuela de cómputo y ciber-café
La escuela de cómputo hasta el momento ha ofrecido cursos en su mayoría por petición de sus clientes, Bases de Internet, Visual Basic, MS Word, MS Excel y elementos básicos de administración de redes.
Asesoría en creación de documentos que resumen investigaciones realizadas en Internet.
Renta de equipos y conexión a Internet
En el corto plazo se tiene pensado explotar el conocimiento en la forma de investigar en Internet haciéndolo por encargo del cliente.

Tabla 5.- Descripción de la UEN Escuela de cómputo y ciber-café.

A continuación se procederá a describir cómo está organizada la empresa con la idea de dar un marco de referencia que ayude a comprender cómo fluye la información, la forma en que se distribuye el trabajo y se toman las decisiones dentro de un proyecto en el que la empresa tiene la administración del mismo.

Organigrama de Enlaces Tecnológicos S.A. de C.V.

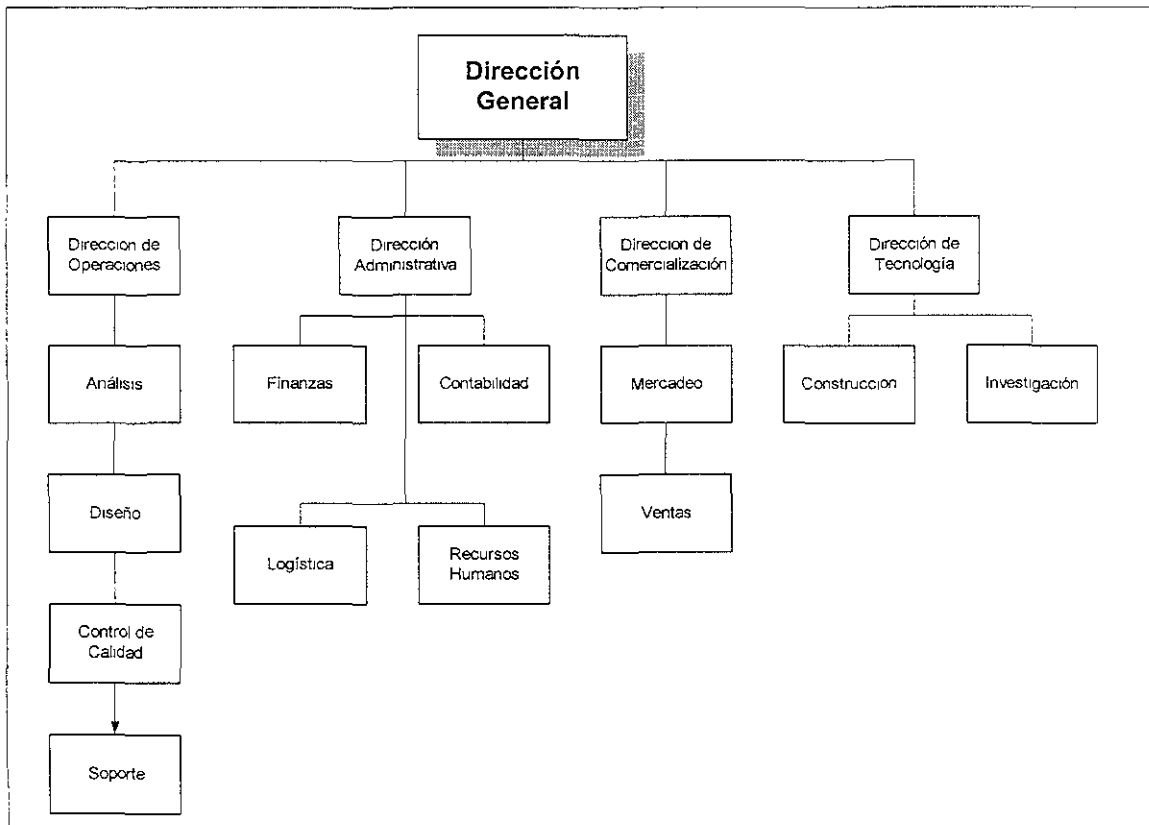


Figura 1.- Organigrama de Enlaces Tecnológicos S.A. de C.V.

Con la idea de describir un proyecto desarrollado a la medida enmarcado dentro de la organización de ET se utilizará un modelo de flujo de información que se muestra en la siguiente imagen:

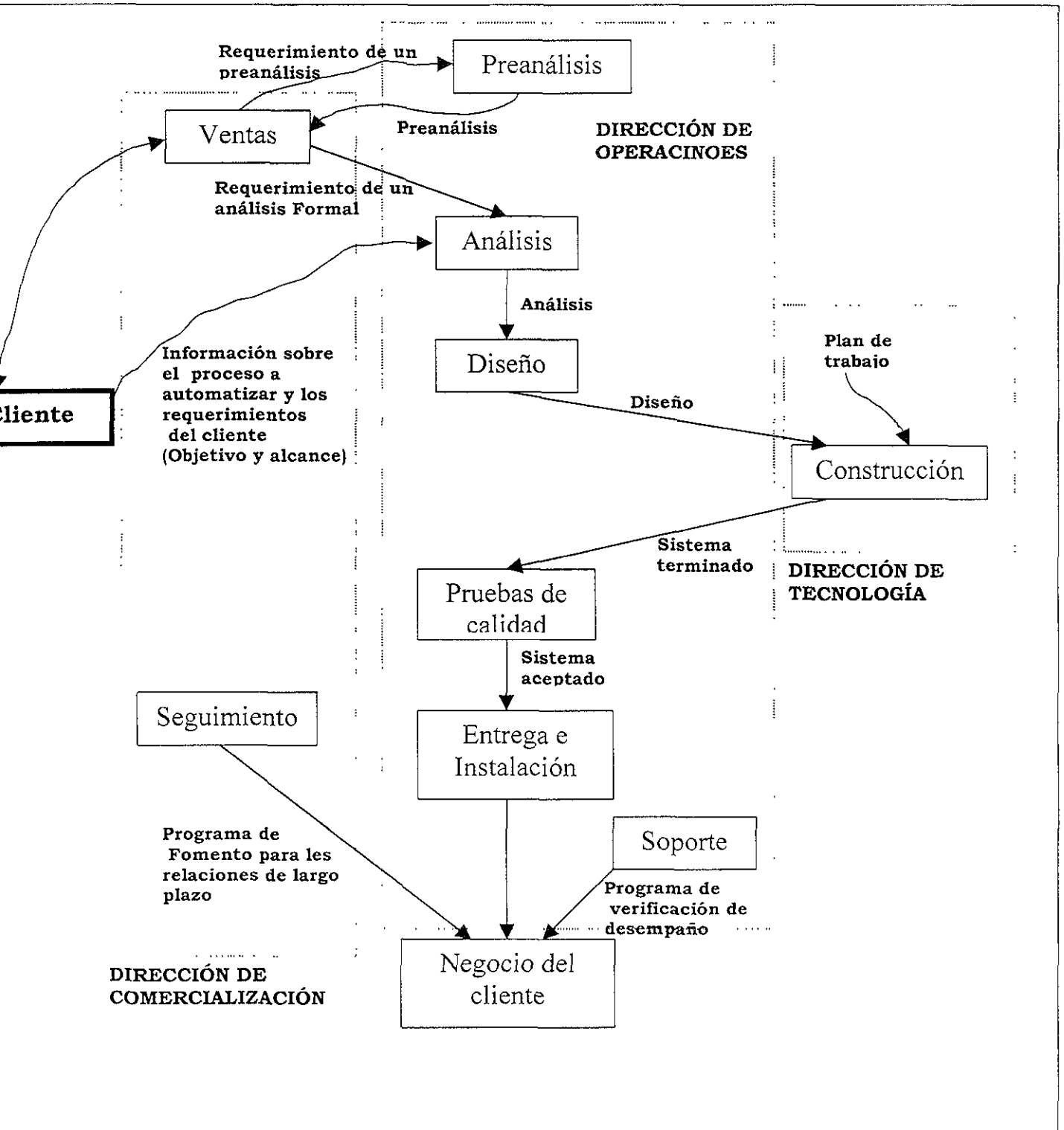


Figura 2 Flujo de información de un proyecto dentro de la organización ET

Distribución geográfica de los negocios de Enlaces Tecnológicos

40 % son realizados en Guadalajara, Jalisco.

50% en la Ciudad de México.

5% en diferentes partes del mundo (Costa Rica, Honduras, Panamá, Colombia y Puerto Rico). Estos últimos han sido proyectos aislados de poca duración.

5% En diferentes partes de la república como Puebla y Coahuila, Veracruz.

Estas cifras fueron tomadas del balance de la empresa para 2000; y actualmente se ha presentado un incremento en los requerimientos para negocios en la Ciudad de México y en el extranjero, particularmente en Colombia.

1.2 Consideraciones sobre empresas similares

El mercado para empresas que brinden soluciones informáticas a la mayoría de los sectores industriales y comerciales, está en crecimiento y por ende existen un sin número de empresas de este tipo, se ha detectado que existen problemas comunes en todas ellas, no se quiere decir con esto que todos los problemas estén presentes en cada una pero sí, que han logrado que el consumidor suela desconfiar de las propuestas de solución que se le ofrecen y por ello prefieran productos estandarizados, por poner algunos ejemplos se tienen los paquetes Microsoft (office) y SAP/R3 entre otros, que aunque tal vez no sean adecuados del todo a sus necesidades sí les infunden la confianza de que no se presentarán problemas posteriores o que existe una empresa importante que los respalde.

Así que con el propósito de identificar los problemas comunes se realizó una encuesta entre diferentes consultores de la industria informática y con clientes que han enfrentado contratiempos con proveedores de software, juntando sus opiniones con la experiencia de los fundadores de la empresa ET, se llegó a resumir los problemas más representativos de los proyectos en la siguiente tabla:

Relacionados con el personal	Relacionados con el proceso	Relacionados con el producto	Relacionados con la tecnología
Motivación débil	Planeación excesivamente optimista	Exceso de requerimientos	Sobreestimación de las ventajas del empleo de nuevas herramientas o métodos
Personal mediocre	Gestión de riesgos insuficiente	Cambio de los requerimientos	Cambio de herramientas a la mitad del proyecto
Fomento de comportamientos heroicos	Fallos de los contratados	Se permiten diseños en exceso meticulosos	Falta de control automático del código fuente
Se suele añadir más personal a un proyecto retrasado	Planeación insuficiente	Se presentan negociaciones en las que las partes se creen enemigos	
Lugares de trabajo inadecuados	Abandono de la planeación bajo presión	Desarrollo orientado mas a la investigación que a la solución	
Fricciones entre los clientes y los desarrolladores	Pérdida de tiempo en el inicio	El sistema es difícil de poner a punto	
Expectativas poco realistas	Se escatima en las actividades iniciales		
Falta de un promotor efectivo del proyecto	Diseño inadecuado		
Falta de participación del usuario	Escatimar en el control de la calidad		
Falta de participación de los implicados	Control insuficiente de la directiva		
Política antes que desarrollo	Convergencia prematura o excesivamente frecuente		
Suposiciones imaginarias para que el plan funcione	Omitir tareas necesarias en la estimación		
	Planear ponerse al día mas adelante		
	Programación a destajo		

Tabla 6.- Problemas recurrentes en empresas proveedoras de software

Explicando con mayor detalle los puntos mencionados en la tabla anterior tenemos:

Relacionados con el personal

- **Motivación débil.**- Se han realizado varios estudios que demuestran que la motivación tiene un efecto importante sobre la productividad y la calidad en el desarrollo de actividades intelectuales (como en el caso del desarrollo de software).
- **Personal mediocre.**- Después de la motivación la capacidad individual de los miembros del equipo, así como sus relaciones como equipo, probablemente tienen la mayor influencia en la productividad.
- **Fomento de comportamientos heroicos.**- Algunos desarrolladores ponen un gran énfasis en la realización de hazañas en los proyectos. Pero lo que hacen tiene más de malo que de bueno porque de esa forma se cae en modelos de planeación *al límite*.
- **Se suele añadir más personal a un proyecto retrasado.**- Éste es quizá el más clásico de los errores que se cometen con frecuencia en el desarrollo de software. Cuando un proyecto se alarga, añadir más gente puede quitar más productividad a los miembros del equipo existente que la que añaden los nuevos integrantes.
- **Lugares de trabajo inadecuados.**- la mayoría de los desarrolladores consideran sus condiciones de trabajo como insatisfactorias. Alrededor del 60% indican que no tienen suficiente silencio ni privacidad. Los trabajadores que están en oficinas silenciosas y privadas tienden a funcionar significativamente mejor que aquellos que ocupan salas ruidosas y repletas.
- **Fricciones entre el cliente y los desarrolladores.**- Se pueden presentar de distintas formas, entre ellas, a los clientes puede parecerles que los desarrolladores no cooperan cuando rehúsan comprometerse con el plan de desarrollo que desean o cuando fallan con entregar lo prometido. A los desarrolladores puede parecerle que los clientes no son razonables porque insisten en planes irreales o cambios en los requerimientos después de que hayan sido fijados. Pueden ser simplemente conflictos de personalidad entre grupos. El principal efecto de esta fricción es la mala comunicación y los efectos secundarios de la mala comunicación incluyen el pobre entendimiento de los requerimientos, pobre diseño de la interfaz del usuario y, en el peor de los casos, el rechazo del cliente a aceptar el producto acabado. En el caso medio, las fricciones entre clientes y desarrolladores de software llegan a ser tan severas que ambas partes consideran la cancelación del proyecto. Para aliviar estas fricciones se necesita tiempo, y distraen tanto a los desarrolladores como a los clientes del trabajo real en el proyecto.
- **Expectativas poco realistas.**- A veces los directivos o los desarrolladores de un proyecto se buscan problemas al pedir fondos

basándose en estimaciones demasiado optimistas, prometen un conjunto de funciones tan altas como la luna, y esto aumenta la fricción entre los involucrados cuando no se cumple lo pactado.

- **Falta de un promotor del proyecto.**- Para soportar muchos de los aspectos del proyecto es necesario un promotor del mismo de alto nivel, para realizar una planeación realista, llevar el control de cambios y la posible introducción de nuevos métodos de desarrollo. Sin un promotor ejecutivo efectivo, el resto del personal de alto nivel de la empresa puede forzar a que se acepten fechas de entrega irreales o hacer cambios que debiliten el proyecto. El consultor australiano (Rob Thomsett afirma que la falta de un promotor efectivo garantiza virtualmente el fracaso de un proyecto (Thomsett 1995).
- **Falta de participación del usuario.**- Cuando no se involucra al usuario en proyectos informáticos, es común que los requerimientos sean mal entendidos y por tanto se vuelven vulnerables a que se consuma tiempo en la realización de prestaciones que mas tarde no se ocuparán.
- **Falta de participación de los implicados.**- La cooperación estrecha sólo se produce si se han involucrado a todos los relacionados en el proyecto para coordinar todos los esfuerzos.
- **Política antes que desarrollo.**- Se ha visto que si existen cuatro equipos hay cuatro tipos de orientaciones olíticas. Los *políticos* están especializados en la *gestión* centrándose en las relaciones con sus directivos. Los *investigadores* se centran en explorar y reunir información. Los *aislacionistas* están solos, creando fronteras para el proyecto que mantienen cerradas a los que no son miembros del equipo. Los *generalistas* hacen un poco de todo: establecen relaciones con sus directivos, realizan investigaciones y exploran actividades, y se coordinan con otros equipos como parte de su modo de trabajo. En un principio los equipos políticos y generalistas están bien vistos por los directivos de lato nivel . Pero después de un año y medio, los equipos políticos llegan a la muerte súbita. Primar la política en vez de los resultados es fatal para el desarrollo que se pretende entregar a tiempo.
- **Suposiciones imaginarias para que el plan funcione.**- Es impresionante cuántos problemas del desarrollo de software se deben a la imaginación. Es común escuchar frases como las siguientes a distintas personas:
 - *Ninguno de los miembros del proyecto cree que pueda completarse el proyecto de acuerdo con el plan que se tiene, pero piensan que quizás si trabajan duro, y **nada va mal**, y tienen un poco de suerte, serán capaces de concluir con éxito.*
 - *Nuestro equipo no hace mucho trabajo para la coordinación de interfaces entre las distintas partes del producto, pero tenemos una*

buena comunicación para otras cosas, y las interfaces son relativamente simples, así que **probablemente sólo necesitemos un día o dos** para eliminar los errores.

- Sabemos que contamos con un desarrollador externo de poco talento para el subsistema de la base de datos, y que es difícil saber cómo va acabar el trabajo con los niveles de personal que ha especificado en su propuesta. No tiene tanta experiencia como algunos de los demás desarrolladores externos, pero puede que compensen con energía lo que les falta en experiencia. **Probablemente acaben a tiempo!**
- No necesitamos reflejar la última lista de cambios en el prototipo para el cliente. **Estoy seguro de que por ahora sabemos lo que quiere.**
- El equipo está diciendo que realizará un esfuerzo extraordinario para cumplir con la fecha de entrega, y que no han legado a su primer hito por pocos días, pero creo que alcanzarán éste a tiempo.

Relacionados con el proceso

- **Planeación excesivamente optimista.**- Fijar un plan excesivamente optimista fomenta que el proyecto falle al infravalorar el alcance del proyecto, minando la planeación efectiva, y reduciendo las actividades críticas para el desarrollo, como el análisis de requerimientos o diseño. También supone una presión excesiva para los desarrolladores, quienes a largo plazo se ven afectados en su moral y su productividad.
- **Gestión de riesgos insuficiente.**- Si no se ejerce una gestión activa de los riesgos, con que sólo vaya mal una cosa se pasará de tener un desarrollo que puede entregarse a tiempo a uno que se entregará atrasado. El fallo de no gestionar uno solo de estos riesgos es un problema común.
- **Fallos de los contratados.**- Las compañías a veces contratan la realización de partes de un proyecto cuando tienen demasiada prisa para hacer el trabajo en casa. Pero los contratados frecuentemente entregan su trabajo tarde, con una calidad inaceptable o que falla al no coincidir con la especificación. Si las relaciones con los contratados no se gestionan cuidadosamente, la utilización de desarrolladores externos puede detener el proyecto en vez de acelerarlo.
- **Planeación insuficiente.**- Si no se planea para conseguir un desarrollo que se entregará a tiempo, no se puede esperar obtenerlo.
- **Abandono del plan bajo presión.**- Los equipos de desarrollo hacen planes y rutinariamente los abandonan cuando se tropiezan con un problema en la planeación. El problema no está en el abandono del plan, sino más bien en fallar al no crear un plan alternativo, y caer entonces en el modo de trabajo de codificar y corregir.

- **Pérdida de tiempo en el inicio difuso.**- El *inicio difuso* es el tiempo que transcurre antes de que inicie el proyecto; este tiempo normalmente se pierde en el proceso de aprobar y hacer el presupuesto. No es poco común que en un proyecto se desperdicien meses o años en un inicio difuso, y entonces se está a las puertas de un plan agresivo. Es mucho más fácil y barato y menos arriesgado suprimir unas pocas semanas o meses del inicio difuso en vez de comprimir el plan de desarrollo en ese tiempo.
- **Escatimar en actividades iniciales.**- Los proyectos que se aceleran intentando acortar las actividades esenciales, y puesto que el análisis de requerimientos, la arquitectura y el diseño no producen código directamente, son los candidatos fáciles. Los proyectos que normalmente escatiman en sus actividades iniciales tendrán que hacer ese trabajo en otro momento, con un costo de diez a cien veces mayor que el que hubieran realizado estas actividades desde el principio.
- **Diseño inadecuado.**- Una caso particular de escatimar en las actividades iniciales es el diseño inadecuado. Proyectos acelerados generan un diseño indeterminado, no asignando suficiente tiempo para él y originando un entorno de alta presión que hace difícil la posibilidad de considerar alternativas en el diseño. El énfasis en el diseño está más orientado a la conveniencia que a la calidad, por lo que necesitará varios ciclos de diseño antes de poder finalizar completamente el sistema.
- **Escatimar en el control de calidad.**- En los proyectos que se hacen con prisa se suele cortar por lo sano, eliminando las revisiones del diseño y del código, eliminando la planeación de las pruebas y realizando sólo pruebas superficiales. Acortar un día las actividades de control de calidad al comienzo del proyecto probablemente supondrá de tres a diez días de actividades finales.
- **Control insuficiente de la directiva.**- A veces no se lleva un control que permita detectar a tiempo los signos de posibles retrasos en el plan, y los pocos controles definidos (si es que los hay) al comienzo se abandonan cuando comienza a haber problemas. Antes de encarrilar un proyecto, en primer lugar se debe ser capaz de decidir si va por buen camino.
- **Convergencia prematura o excesivamente frecuente.**- Bastante antes de que se haya programado entregar un producto, hay un impulso para prepararlo para su entrega, mejorar su rendimiento, imprimir a documentación final, incorporar entradas en sistema final de ayuda, pulir el programa de instalación, eliminar las funciones que no van a estar listas a tiempo y demás. En proyectos hechos con prisa hay una tendencia a forzar prematuramente la convergencia. Puesto que no es posible forzar esta convergencia todos los intentos por llevarla a cabo son sólo una pérdida de recursos.

- **Omitir tareas necesarias en la estimación.**- Si la gente no guarda cuidadosamente datos de proyectos anteriores, olvida las tareas menos visibles, pero son tareas que se han de añadir.
- **Planear ponerse al día más adelante.**- Cuando existe retraso con respecto al plan de un proyecto y ya se ha convivido con él suficiente tiempo se está en la posibilidad de hacer una reestimación del tiempo que será necesario para terminar el proyecto, es muy común que en lugar de hacer esta reestimación se planee recuperar el tiempo mas adelante, lo que por lo general lleva a un retraso todavía mayor que el que se tiene.
- **Programación a destajo.**- Algunas organizaciones creen que la codificación rápida, libre, tal como salga, es el camino hacia el desarrollo a tiempo. Piensan que si los desarrolladores están lo suficientemente motivados, podrán solventar cualquier obstáculo. Esto está, en la mayoría de los casos, lejos de ser verdad. Es en realidad la envoltura del viejo paradigma a destajo combinado con una planeación ambiciosa, y esta combinación ha fallado en repetidas ocasiones.

Relacionados con el producto

- **Exceso de requerimientos.**- Algunos productos tienen más requerimientos que los que necesitan, desde el mismo inicio. La eficiencia se fija como requisito más a menudo que lo que es necesario, y puede generar una planeación del sistema innecesariamente larga. Los usuarios tienden a interesarse menos en las prestaciones complejas que en las de las secciones de mercadotecnia o de desarrollo, y las prestaciones complejas alargan desproporcionadamente el plan de desarrollo.
- **Cambio de los requerimientos.**- Incluso si se han evitado los requerimientos excesivos los proyectos sufren como media un 25% de cambios a lo largo de su vida.
- **Se permiten diseños en exceso meticulosos.**- Los desarrolladores encuentran fascinante la nueva tecnología por lo que hacen diseños que involucran técnicas o herramientas que no dominan del todo y eso en general retrasa los proyectos.
- **Se presentan negociaciones en las que las partes se creen enemigos.**- Cuando un directivo aprueba un retraso en el plan de un proyecto que progresa mas lento de lo esperado, y entonces añade tareas completamente nuevas después de un cambio en el plan (acción muy frecuente en proyectos de gran importancia y magnitud), se produce una situación curiosa. La razón subyacente de esto es difícil de localizar, puesto que el directivo que aprueba el retraso lo hace implícitamente sabiendo que el plan estaba equivocado. Pero una vez que se corrige, la misma persona realiza acciones explícitas para volver a equivocarse. Esto sólo puede ir

en contra del plan y en la moral del equipo porque de ese momento en adelante los desarrolladores sentirán que no trabajan en equipo con la dirección del proyecto.

- **Desarrollo orientado más a la investigación que a la solución.-** Si el proyecto fuerza los límites de la informática porque necesita la creación de nuevos algoritmos o de nuevas técnicas de cómputo, no se está desarrollando software; se está haciendo investigación de software. Los planes de desarrollo de sistemas informáticos es razonablemente predecible, los planes de investigación ni siquiera son predecibles teóricamente.
- **El sistema es difícil de poner a punto.** Muchos sistemas que se terminan no son instalados porque resulta muy complicado migrar los procedimientos que la empresa día con día realiza de una tecnología a otra sin dejar de operar.

Relacionados con la tecnología

- **Sobreestimación de las ventajas del empleo de nuevas herramientas o métodos.-** Las organizaciones mejoran raras veces a grandes saltos, sin importar cuantas nuevas herramientas o métodos empleen o lo buenos que sean. Los beneficios de las nuevas técnicas son parcialmente desplazados por la curva de aprendizaje. Las nuevas técnicas también suponen nuevos riesgos que sólo se descubren al utilizarlas.
- **Cambio de herramientas a mitad del proyecto.-** Es un viejo truco que funciona raramente. A veces puede tener sentido actualizar incrementalmente dentro de la misma línea de productos, de la versión 3 a la versión 3.1 por ejemplo. Pero cuando se está a la mitad de un proyecto, la curva de aprendizaje, rehacer el trabajo, y los inevitables errores cometidos con una herramienta totalmente nueva, normalmente anulan cualquier beneficio posible.
- **Falta de control automático del código fuente.-** Esto expone a los proyectos a riesgos innecesarios. Sin él, si dos desarrolladores están trabajando en la misma parte del programa, deben coordinar su trabajo manualmente. Deberán ponerse de acuerdo para poner la última versión de cada archivo en el directorio maestro y verificarlos con los demás antes de copiarlas en este directorio. Pero en general se sobrescribe del otro. Se desarrolla nuevo código con interfaces desfasadas, y después se tiene que rediseñar el código al descubrir que se ha utilizado una versión equivocada de la interfaz. Los usuarios avisan de errores que es imposible reproducir porque no hay forma de volver a crear los elementos que han utilizado.

Según la experiencia que los diferentes participantes en el estudio los problemas más impactantes relacionados con el personal son: Expectativas poco realistas, Falta de participación de los implicados, suposiciones imaginarias para que el plan funcione y el fomento de comportamientos heroicos (estos puntos están marcados con negritas en la tabla para remarcar su importancia); relacionados con el proceso: Diseño inadecuado, planeación excesivamente optimista, gestión de riesgos insuficiente, abandono de la planeación bajo presión, pérdida de tiempo en el inicio y omitir tareas necesarias en la estimación; relacionados con el producto: Cambio de los requerimientos y que el sistema es difícil de poner a punto, por último los relacionados con la tecnología: Sobreestimación de las ventajas del empleo de nuevas herramientas o métodos y la falta de control automático del código fuente. Así que como una medida adecuada se deben poner los efectos de este tipo de problemas como riesgos y hacer esfuerzos importantes para evitarlos.

A continuación se hace mención al entorno de competencia a que se enfrentan las empresas que ofrecen soluciones informáticas, se debe hacer énfasis en que sobre todo las empresas de reciente creación son más vulnerables.

1.3 Problemática de mercado

Los requerimientos del mercado han cambiado a raíz de los problemas mencionados en la tabla anterior, ahora las empresa consumidoras de estos servicios exigen no sólo que se les garantice el producto sino que también quieren conocer el proceso para poder conocer a la mayor brevedad si existen desviaciones con respecto de lo prometido, en la siguiente figura se muestra la forma en que se encuentra el entorno de las empresas proveedoras de soluciones informáticas y en particular de Enlaces Tecnológicos.

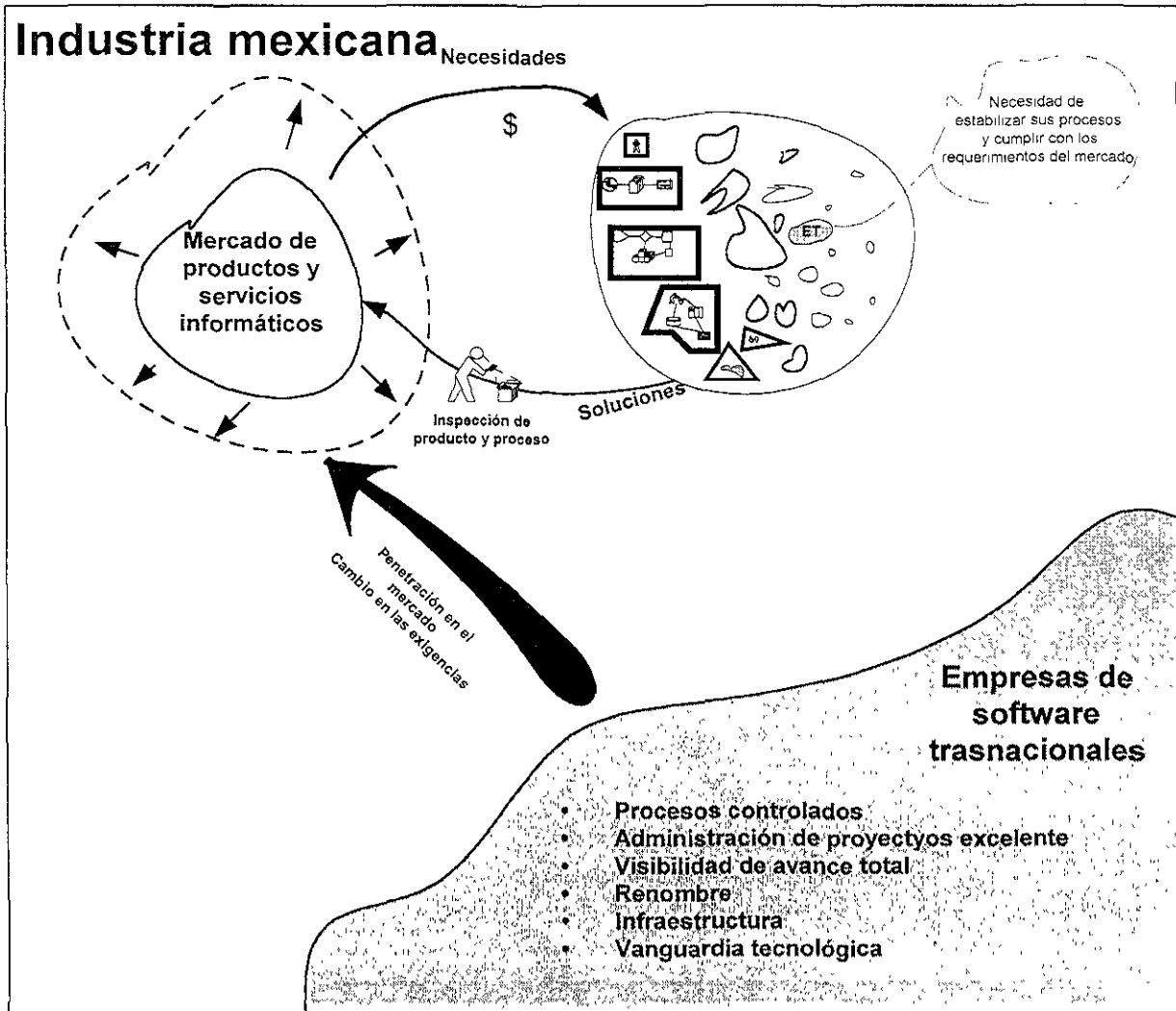


Figura 2.- Entorno de las empresas proveedoras de software

Por otra parte el mercado de productos y servicios informáticos es de cambios vertiginosos, porque la tecnología en esta sentido evoluciona a gran velocidad tanto en el software como en el hardware, lo que hace que las necesidades de los clientes varíen constantemente; nuevos equipos pueden ofrecer al cliente mayores beneficios lo mismo que los programas pero es común que las prestaciones que son ofrecidas a los usuarios requieran de equipos más potentes.

Una vez que se ha identificado al mercado de productos y servicios informáticos como de *movimiento rápido* se debe pensar en que las organizaciones estarán obligadas a desarrollar estrategias con dos dimensiones importantes en consideración: la flexibilidad y la habilidad de controlar sus procesos.

Las compañías proveedoras de software deberán operar con la suficiente flexibilidad para poder adaptarse a rápidos movimientos en el mercado y con la habilidad de controlar sus procesos, la forma de empatar estas dos variables se logra definiendo la flexibilidad como una tarea administrativa que se encargará de la habilidad de la compañía para tener control; Y la definición de tareas dentro de la organización para adaptarse a diferentes condiciones.

Considerando las dos dimensiones que se han mencionado se puede clasificar a las empresas en cuatro categorías básicas:

- Rígidas.- Tiene una pobre mezcla de flexibilidad y control, o la habilidad de cambiar es lenta. Se caracteriza porque sus operaciones son básicamente rutinas simples, además sus opciones de cambio son limitadas, la improvisación se considera un tabú, en general presentan una estructura centralizada con muchas leyes jerárquicas que junto con una cultura de *cero iniciativas intelectuales* reducen el potencial de la empresa para la flexibilidad y las vuelven frágiles y vulnerables.
- Planeadas.- Estas organizaciones también tienen problemas de flexibilidad pero la variedad en las rutinas y la habilidad de controlar sus procesos están menos limitadas que en la organizaciones rígidas. La flexibilidad proviene de reglas específicas y procesos detallados que requieren un gran cantidad de información procesada. Sin embargo, para cada cambio posible la organización debe desarrollar ciertas rutinas.
- Flexibles.- Se caracterizan por tener estrategias y estructuras flexibles, con una habilidad de controlar sus procesos razonablemente alta. Con esto los disturbios en el mercado se pueden afrontar efectivamente puesto que los cambios se pueden implementar en poco tiempo con estructuras de *no rutina*.

- **Caóticas.**- Son organizaciones altamente flexibles aunque totalmente incontrolables.

En la siguiente gráfica se muestra una tipología de situaciones de empresas en lo relativo a la habilidad para controlar sus procesos y a la flexibilidad (o rigidez) de sus estructuras, elementos importantes en entornos como el tecnológico que presentan condiciones de movimiento rápido.

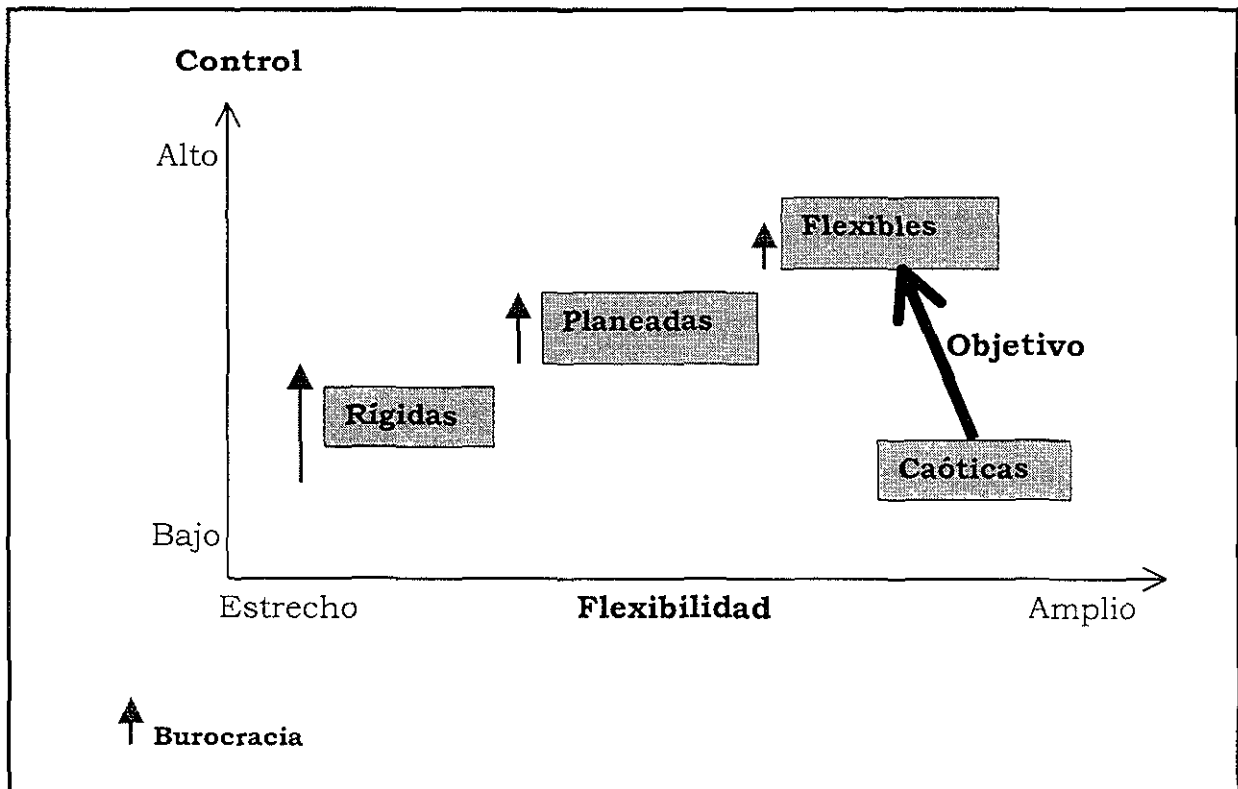


Figura 5 Tratando con la turbulencia de mercados de cambio rápido

Building flexible Organizations for fast-Moving Markets

Henk W. Volberda

Long Range Planning, Vol. 30, No. 2, pp196-183 (1997)

Una vez que se ha explicado qué es ET, cómo está organizada, y el entorno en el que se desenvuelve, se procederá a definir los objetivos que persigue este trabajo.

1.4 Objetivo

Diseñar un procedimiento para estabilizar el desarrollo de los proyectos informáticos que le son encargados a la empresa, esto es, diseñar un formato con directrices y procedimientos sencillos, comunes y sensatos, expresados en un lenguaje comúnmente entendido: directrices y procedimientos que establezcan un puente entre las diferencias que existan dentro de un equipo de trabajo y sus funciones individuales; directrices y procedimientos que el equipo pueda usar mancomunadamente para cumplir con sus responsabilidades sin inhibir las aportaciones personales pero sin añadir nuevas tareas innecesarias, es decir, que no generen burocracia.

Los objetivos también se pueden agrupar en los siguientes tres tipos:

Táctico-Operacionales
Garantizar que los proyectos se entreguen a tiempo con la calidad y el costo estipulado.
Especificar las responsabilidades de cada participante en el proyecto.
Formar la memoria de referencia para futuras estimaciones.
Reutilización de código.
Formulación de las bases de la gestión de riesgos en la planeación de los proyectos.
Normalizar la conceptualización del ciclo de vida del proyecto.
Organizar la comunicación entre el equipo de trabajo y el cliente.
Sentar las bases para una cultura de Mejora Continua en la empresa.
Facilitar la toma de decisiones descentralizada de los equipos de trabajo y las actividades en paralelo.

Tabla 7.- Objetivos táctico-Operacionales.

Estratégicos
Cumplir con un requisito del mercado.
Hacer del proceso de administración de proyectos informáticos en la empresa un pilar para la diferenciación de la compañía (El entendimiento que se tiene del problema, lo claro que resulta la solución propuesta y la efectividad del plan entre otros)
Iniciar los esfuerzos de estandarización para una futura certificación que dará una ventaja a la empresa en las diferentes competencias en las que participa.
Contribuir a suministrar información relevante para determinar cuál debe ser el plan de negocios para una propuesta específica.

Tabla 8.- Objetivos estratégicos

Prospectivos
Demostrar que la empresa posee un sistema <i>adaptado</i> para responder a solicitudes que no se anticipen en entornos impredecibles.
Lograr que los participantes de la empresa se sientan orgullosos de la forma en que la empresa lleva sus proyectos y se comprometan a mejorar ésta cada día.
Como fuente de retroalimentación de los cambios que se presenten en el futuro

Tabla 9.- Objetivos Prospectivos.

Con base en lo expuesto anteriormente y con pláticas informales en las que se encontró que la empresa funcionaba, a pesar del éxito hasta el momento obtenido, con deficiencias en los tiempos de respuesta a una solicitud de presupuestos para proyectos, en el control de los proyectos y en la formalización de procedimientos entre otros y con el sentimiento patente de que existía un malestar en los equipos de trabajo por razones que no eran evidentes se procedió a tener un primer acercamiento con personal de la empresa junto con algunos directivos, la técnica que se utilizó fue la TKJ y a continuación se describe en un diagrama.

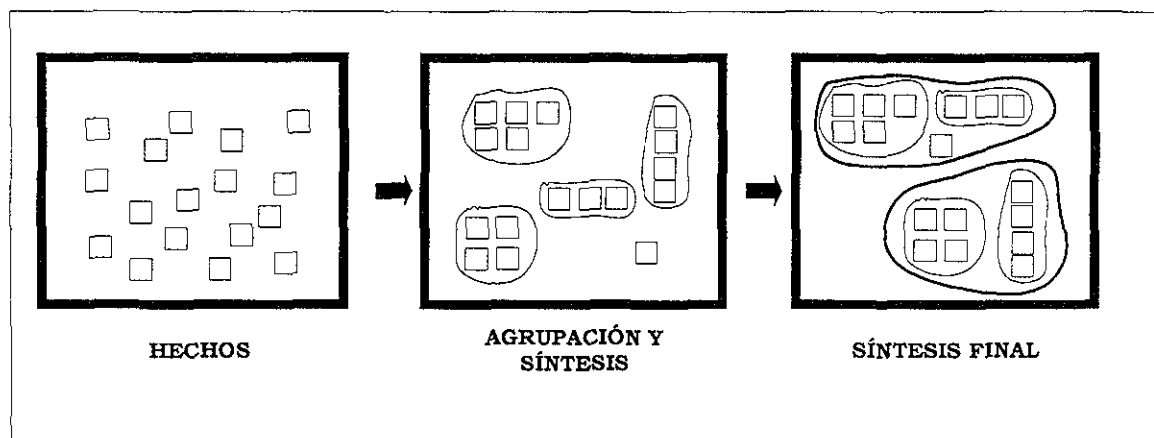


Figura 6 Diagrama de la técnica TKJ

Obtenido del libro
Técnicas para el análisis de sistemas parte 1
Dr. Gabriel Sánchez Guerrero
Segunda edición
División de Estudios de Posgrado, Facultad de Ingeniería UNAM

Los resultados de este ejercicio se presentan en el punto 1.5 que a continuación se muestra.

1.5 Análisis de la empresa

Dinámica para el análisis (TKJ)

Organización de equipos de para realizar algunas sesiones de trabajo

En un primer acercamiento con la empresa y sus colaboradores en conjunto se realizó una reunión en la que se aplicó la técnica TKJ para hacer un diagnóstico general del desempeño de la empresa, se detectaron varias áreas en las que se puede hacer mejoras y algunas en las que vale la pena elaborar proyectos de mayor importancia pero que por el tiempo programado para este trabajo y por el interés mismo de la empresa se han dejado para un mediano plazo, a continuación se presenta el ejercicio TKJ aplicado el 15 de Octubre de 1999.

Los participantes fueron cinco personas:

- Director General (DG)
- Director de Tecnología (DT)
- Consultor de sistemas C1)
- Consultor de sistemas (C2)
- Consultor de sistemas (C3)

El problema genérico se definió como "***estancamiento en los planes de desarrollo de la empresa.***"

Formulación del problema

1. Hace poco surgió una duda y no supimos a quién consultar (del personal de ET) (C3)
2. Arturo, Luis, Miguel, Héctor, Fernando y los demás no saben C C++ Java UNIX ni PERL (DT)
3. Rentamos un departamento para hacer un laboratorio y actualmente es una bodega en la que cuesta trabajo entrar (DG)
4. No se animó la empresa a "entrarle" al proyecto de la Secretaría de Saludo
5. Juan Gabriel nos dejó colgados y quedamos mal en un proyecto (DT)
6. El proyecto TBANC va atrasado (DT)
7. Hace n meses que no tomo vacaciones, donde n es un entero positivo mayor que 24 (DT)
8. Arturo no sabe usar el programa de contabilidad para revisar los números de la empresa (DG)
9. Regularmente no se la da seguimiento a una idea o plan como en el caso Citybanc (C2)
- 10.El curriculum de la empresa no está bien (el documento que utilizamos para entregar, no el historial de la empresa (DT)

11. Nos llegó un correo electrónico para solicitar apoyo en Australia y lo Arturo y César lo contestaron un mes y medio después (DG)
12. Fernando y Héctor no dominan CITOS (DG)
13. Los cursos no se planean (C2)
14. Arturo no tiene una LAPTOP y por eso no le puedo enviar correo seguro (DT)
15. En Banamex le dio miedo a quemarse a la administración asignar gente. (C1)
16. El sistema de Listas de Rayas no se ha vendido (DT)
17. Arturo pidió a Rogelio un informe financiero (por mes) y jamás lo ha revisado (DT)
18. La intranet de ET tiene deficiencias conocidas que no se han reparado, ni se tiene un plan para hacerlo (DT)
19. Hace ocho días no encontré clips, ligas ni sobres en ET (DG)
20. Para la gente de Banamex no es muy claro lo qué hago (C2)
21. Fernando no está enterado de todos los problemas (C1)
22. NO se delegan ñas responsabilidades adecuadamente (arturo Luis y Miguel) (C3)
23. Se tiene que estar en Serfin para hacer trabajo sin trabajar (C2)
24. Arturo y Luis cuando se les platica algo nuevo, dicen OK después lo vemos (C2)
25. Luis Zinker está viviendo en Máxico DF (C2)
26. La oficina de Guadalajara sale a la Internet por un modem (DT)
27. Hubo vacilación para realizar la presentación a Citybanc, hasta el momento según se no se ha hecho nada (C1)
28. No hay comunicación entre el personal de México y el de Guadalajara (C3)
29. Hoy se fue la luz y no pudimos terminar la tarea en Pino (DG)
30. No tengo credencial de Enlaces (DT), He dormido 4 horas diarias por dos semanas (al menos) por atender problemas de dos proyectos (DT)
31. Serfin emitió un cheque para ET en Mayo y Luis no lo recogió hasta Junio (27 días después) (DG)
32. En febrero comparamos equipo para estaciones LINUX y hoy están apiladas en la sala de juntas y en la bodega (DG)
33. Mi esposa está enojada, no la he visto en dos meses mas que cuatro días (DT)
34. El WEBSITE de la empresa no se ha actualizado desde que lo publiqué (DT)
35. La impresora láser aún no está conectada al hidalgo (DG)
36. No se puede tomar un nuevo proyecto porque ya están involucrados en alguno otro César, Arturo, Luis y Miguel que son por quienes normalmente se inclinan a contratar los clientes (C1)
37. Hemos perdido mucho tiempo en juntas en Banamex (DT)
38. Dejé a Israel mucho tiempo sin trabajo (DT)

Se agruparon en sobres quedando de la siguiente forma:

A.- Hay actividades necesarias no asignadas:
9, 10, 11, 16, 17, 27

B.- La logística de la empresa está desorganizada
3,14, 19, 29, 30

C.- Nadie se ocupa de los sistemas internos de la empresa:
8, 18, 26, 32, 34, 35

D.- Falta de criterios establecidos para la toma de decisiones
4,15

E.- Falta de capacitación
2, 12, 13

F.- Falta de comunicación
1, 20, 21, 28

G.- Sobre utilización de recursos (de personal) con habilidades conocidas)
7,25, 30, 33, 36

H.- No se cuenta con un plan de seguimiento adecuado para el trabajo de los consultores
5,6

I.- Mal manejo de la gente
22, 23, 25, 31, 37, 38

Posteriormente se agruparon en 6 categorías que hubieran podido reducirse pero se decidió que eso generalizaba problemas específicos y detallados

- I. Falta de capacitación (E)
- II. Desorganización en la logística (B)
- III. Falta de comunicación (F)
- IV. Actividades necesarias no asignadas (A, C)
- V. Falta de definición de políticas operativas (D, H)
- VI. Carencia de un esquema que defina responsabilidades y un plan de asignación de actividades (G, I)

Se realizó una etapa final para agrupar quedando de la siguiente forma:

- Carencia de procedimientos definidos para la asignación, definición y seguimiento de responsabilidades y para los canales de comunicación interna. [III, IV, V y VI]
- Falta de un esquema de capacitación.
- Desorganización en la logística.

Finalmente se estableció que la causa raíz del problema es que hay una carencia de un plan de administración de recursos con políticas establecidas que faciliten la comunicación interna.

En reuniones posteriores en las que se trató de identificar las posibles acciones que llevaran a mejorar los problemas mencionados se acordó dar como primer paso la creación de un **procedimiento para administrar los proyectos en los que interviene la empresa**, que involucre la definición de responsabilidades, la administración de los recursos, la comunicación y el seguimiento entre otros. En esta decisión influyó también el hecho de que perdieron un contrato por no poder presentar la metodología con la que opera la empresa.

A continuación se realizaron pláticas informales con integrantes de la empresa para identificar las inquietudes que tenían con respecto a las implicaciones que el establecimiento de procedimientos para la administración de proyectos.

Las cuestiones más comunes que se encontraron fueron que:

- Las experiencias con metodologías de otras compañías habían llevado siempre a hacer planes utópicos.
- Que los pasos en la administración de un proyecto, lejos de ayudar, generaban burocracia en extremo.
- Que sí era útil pero que en general la metodología dejaba de seguirse cuando la presión aumentaba.
- Por lo general una metodología de este tipo es tan general que no aplica para ningún caso particular o para todos sin brindar ninguna ayuda práctica.
- Y que nadie conoce todos los pasos que la metodología presupone entre otros muchos.

La siguiente actividad que se llevó a cabo fue hacer una investigación sobre la teoría de la administración de proyectos informáticos (que se presenta en los apéndices) y de las prácticas que siguen algunas empresas de este ramo, con el objeto de sintetizar los lineamientos generales para diseñar una forma particular en la que la empresa administrará sus proyectos.

1.6 Identificación de la necesidad de intervenir

Al conocer la situación de la empresa y hacer la síntesis de las inquietudes y objetivos de los participantes de la empresa se podrían formular tres problemas centrales:

- El grado de madurez de la empresa en relación con la administración de proyectos está en una etapa empírica.
- Las condiciones del mercado obligan a las empresas a tener formas de trabajo definidas y normalizadas que permitan a los clientes conocer en todo momento el avance del proyecto.
- Los tiempos de respuesta en la elaboración de propuestas son muy lentos porque se hacen en juntas de los directivos para plantear el criterio con el que se hará.

2. Diseño del procedimiento para la administración de proyectos informáticos en Enlaces Tecnológicos S.A. de C.V. (ET)

A continuación se presenta la forma genérica en que puede operar un procedimiento para administración de proyectos informáticos en empresas que responden a necesidades particulares de los clientes, con la idea de lograr, junto con los integrantes de la empresa, un entendimiento claro de su operación y de ahí partir definir las particularidades que la empresa proponga, para hacerlo suyo desde la fase de concepción y con ello evitar el rechazo que todo cambio impuesto conlleva.

2.1. Definición de especificaciones para el procedimiento de administración de proyectos en Enlaces Tecnológicos.

En la siguiente lista se presentan los aspectos relevantes que se han de definir para que el procedimiento de administración de proyectos opere con base en las inquietudes y experiencia de los participantes de la empresa, y fundamentándose en la teoría de administración de proyectos y observaciones de los intereses de los clientes y la opinión de expertos.

Preanálisis

Se deben identificar cinco aspectos (objetivo, alcance, antecedentes, contactos y procesos guía) para estar en situación de hacer un preanálisis con fundamentos suficientes para que la propuesta de solución que la empresa realizará genere pocas variaciones con respecto de un análisis formal, que ya se realiza con el contrato firmado.

A. Objetivo

- ¿Qué se quiere del sistema?
- ¿Cuál es la finalidad de la información que genera el sistema?
- ¿Cuál es el objetivo del sistema?
- ¿Qué función debe desempeñar?
- Es recomendable escribirlo con las palabras del cliente

B. Alcance

- Identificar fases (si las hay)
- Entregables
- Prioridades

- Fechas límites

C. Antecedentes

- Tamaño de la empresa
- Giro
- Cuál es el equipo con que cuentan, o con qué tecnología se desea que opere
- Con qué arquitectura se realizará
- Con qué equipos o programas debe convivir
- Que políticas y estándares tienen para las soluciones informáticas
- Cómo es la comunicación entre los equipos
- Se tienen licencias para operar productos comerciales de cómputo
- Si se tiene algún presupuesto ya definido

D. Contactos

- Quiénes serán los dueños del sistema (Quiénes utilizan la información generada por el sistema)
- Qué áreas están involucradas
- Personas que tienen injerencia en el proyecto
- Quién representará al cliente durante el proyecto

E. Procesos guía

- Procedimiento actual del sistema a actualizar incluyendo manuales de procedimientos: ¿qué es lo que da origen a la actividad?, ¿cuánto tiempo tarda actualmente cada actividad?, ¿qué factores intervienen en la duración de la actividad?
- Cuales son o serán las entradas y salidas (datos genéricos) y puntos de entrada o área.
- ¿En qué forma se reciben los datos del sistema?, ¿Cómo se deben almacenar?, ¿Durante cuanto tiempo? y ¿cuáles son las políticas de depuración para al almacén de información?
- Identificar salidas (tipos de reportes) y los puntos de salida en donde los requieren.
- Formas de captura y reportes existentes.

2.2 Estipulación de criterios y lineamientos

Con la idea de unificar criterios al elaborar una propuesta hubo que definir diferentes lineamientos sobre algunos aspectos importantes, a saber:

- **Esfuerzo.** Se acordó definir el concepto de *objeto* (no son exactamente iguales que los que se definen utilizando C++); Un objeto representa diferentes cosas en diferentes contextos informáticos, por ejemplo en el caso de un desarrollador de sistemas que lo hace en Pascal será una *función*, para un diseñados de páginas de Internet será una *pantalla*, en un sistema complejo los objetos serán los *reportes* que deban ser entregados, *las pantallas* de interfaz con el usuario. Una vez que se acordó utilizar el concepto de *objeto* (Cantidad promedio de actividades que realiza un desarrollador por unidad de tiempo) se definió el esfuerzo necesario en función del número de objetos del proyecto.
- **Personal asignado al proyecto.** Para asignar el personal al proyecto se decidió utilizar la Metodología para el Análisis de Decisiones que proponen Kepner y Tregoe en su libro *el nuevo directivo racional* con los siguientes criterios:

Objetivos de la decisión (criterios)	Definición	Peso
Obligados	<ul style="list-style-type: none"> • Capacidad técnica • Disponibilidad 	
Deseados	<ul style="list-style-type: none"> • Responsabilidad (Formalidad) • Experiencia en proyectos similares • Capacidad para la toma de decisiones • Actitud (habilidad de autoaprendizaje) • Lealtad a la empresa • Facilidad para relacionarse con los clientes • Conocimiento de la forma de trabajo de ET • Liderazgo 	<ul style="list-style-type: none"> • 10 • 9 • 8 • 7 • 7 • 6 • 5 • 4

Los pesos asignados a los criterios y los criterios mismos pueden variar en casos especiales de proyectos, pero en forma general estos son los que la empresa utilizara para desempatar cualquier decisión sobre el personal que se asignará a los proyectos

Organización del equipo de trabajo Genérica

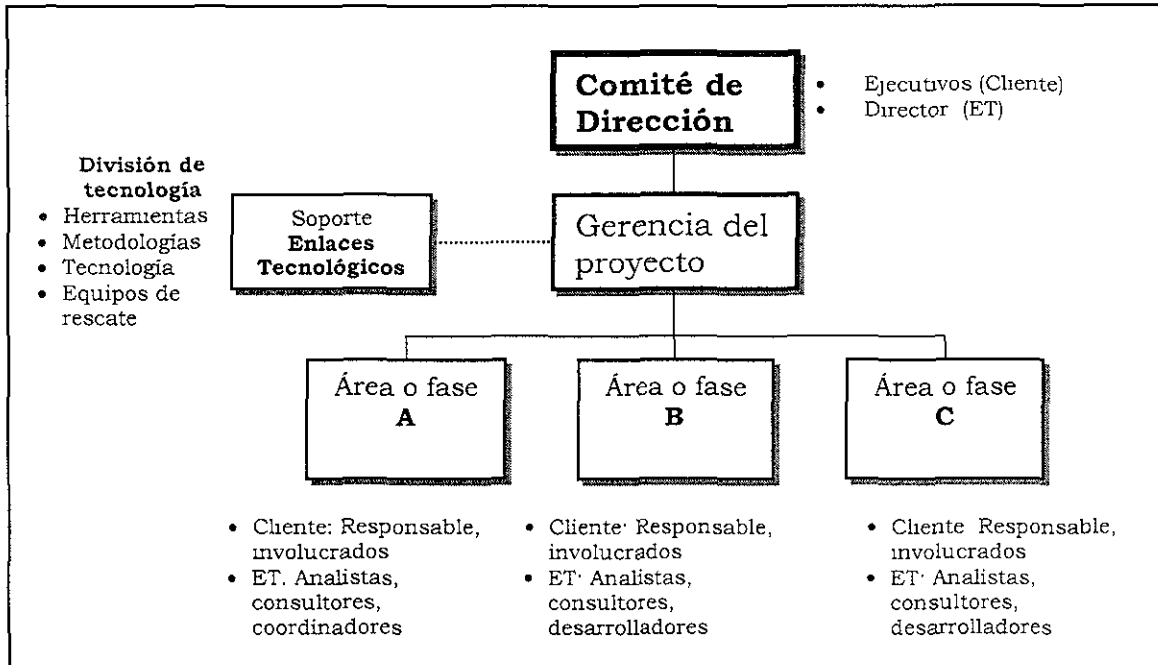


Figura 7 Organigrama de los equipos de trabajo en proyectos a la medida

- **Tiempo.** Una vez que se sabe cuál es el esfuerzo necesario y de qué personal se dispondrá para desarrollar el sistema el cálculo del tiempo es simple; pero se tuvo que definir cuál era el margen de seguridad que en la empresa se contemplará en los proyectos futuros, el algoritmo de cálculo tiene contemplados aspectos como el volumen (número de objetos similares), motivación, acceso a información y capacidad técnica. Con esto se diseñó un algoritmo de cálculo y se probó contra la experiencia de la empresa con otros proyectos y fue aceptado como un estándar en la empresa para calcular el tiempo que tardarán en desarrollar un sistema.

$$F(x) = \begin{cases} 2x + 10 & ; 1 \leq x \leq 10 \\ x + 20 & ; 10 < x \leq 30 \\ (\frac{1}{2})x + 35 & ; 30 < x \leq \infty \end{cases}$$

Donde x es el número de objetos y la variable y representa el tiempo en que se terminará el sistema.

Por otro lado se acordó tomar en cuenta un factor subjetivo que puede hacer cambiar el tiempo de desarrollo, la motivación (que se tratará un poco mas adelante); una persona motivada puede reducir el tiempo promedio de desarrollo a la mitad y en contraparte una persona desmotivada puede hacer que el tiempo se duplique el responsable del proyecto deberá conocer el estado de motivación de las personas que trabajarán en el proyecto para hacer más confiable su estimación, por lo que en la gráfica se presentan tres curvas la serie 2 hace referencia a personas con baja motivación por el proyecto, la serie 1 a personas con motivación normal y la serie 3 con alta motivación.

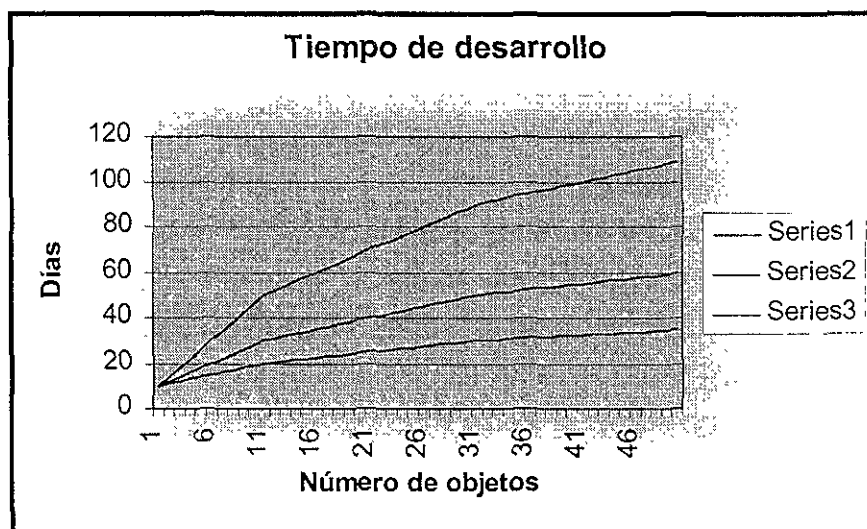


Figura 8 Gráfica del desempeño en función de la motivación

Al haber definido la estimación del tiempo en función de los objetos y con la ayuda de la estructura de costos y políticas para fijar el margen de utilidad de la empresa es posible hacer una propuesta con base en un presupuesto fijo o con un tiempo acotado e inferior al que la empresa estimó que se tardaría en desarrollar el sistema utilizando las funciones inversas.

Por otro lado es importante hacer algún comentario sobre la motivación, así que en el apéndice 1 sobre los elementos de la administración de proyectos se expondrán algunas consideraciones.

- **Costos.** Se definió una estructura para calcular cómo hará la compañía para conocer sus costos directos (sueldos) y sobretodo los indirectos (administrativos, de equipo, de capacitación y el costo de oportunidad).

- **Margen de utilidad.** Fue necesario definir la política que la empresa habrá de seguir para calcular el margen de utilidad *base* (100% del costo) y los aspectos que lo modificarán según haya variantes en los proyectos; los aspectos que modifican el margen de utilidad son:

Aspecto	Categorías	Porcentaje de modificación del margen de utilidad
Tamaño del proyecto	Pequeño	Sin modificación
	Mediano	-5%
	Grande	-20%
	Gigante	-50%
Posibilidad de hacer más negocios con el cliente	Alta	-30%
	Baja	+30%
Experiencia de la compañía	Nula	-20%
	Baja	-10%
	Media	Sin modificación
	Alta	+20%
	Única*	+30%
Disponibilidad de los recursos que realizarán el sistema	Baja	+40%
	Media	+20%
	Alta	Sin modificación
Lugar de desarrollo	Enlaces Tecnológicos	Sin modificación
	Local	+5%
	GUA., DF ó 2ª Patria**	+10%
	Resto de la república	+15%
	Extranjero	+20%
Interés	Nulo	+2000%
	Bajo	+20%
	Medio	Sin modificación
	Alto	-10%
Riesgo de atraso en el pago	Alto	+10%
	Bajo	Sin modificación

* Con único se debe entender que la probabilidad de encontrar una persona fuera de ET es muy baja

** Son ciudades en las que los participantes en el proyecto han vivido, o tienen familia o se sienten como en casa.

Así pues el margen de utilidad se calcula como la suma de los costos mensuales de cada una de las personas que participará en el proyecto multiplicada por el porcentaje de utilidad base modificado por los aspectos que se describen en la tabla.

2.3 Formato de propuesta

- Se definió:
 - Papel en que se presenta: Membretado con el logotipo de Enlaces Tecnológicos en la parte superior y las direcciones (Guadalajara y México D.F.) en la parte inferior izquierda y la de correo electrónico en la parte inferior derecha.
 - Tipo de letra: Arial tamaño 12, excepto en el título del documento en el que será de tamaño 20, escrito con mayúsculas y con negritas.
 - Estilo del documento:
 - En la portada: El nombre de la empresa a la que se dirige el documento (Nombre y dirección), atención a la persona que va dirigida, Objetivo del documento. Nota sobre los cambios en el sistema cuando se conozcan con mayor profundidad los requerimientos del sistema (es decir, después del análisis detallado), nota de agradecimiento por permitir a la empresa presentar la propuesta y nombre, cargo y firma del responsable del proyecto por parte de Enlaces Tecnológicos.
 - En el contenido: los títulos realizados con *negritas* y separados dos renglones de su explicación, las explicaciones justificadas en la hoja y las hojas deben ir numeradas.
 - Orden en que se presentará el contenido:
- Los aspectos que debe abordar el contenido y el orden en el que se presentarán son básicamente los siguientes:
 - Portada
 - Nombre del proyecto
 - Nota sobre la confidencialidad.
 - Índice del contenido.
 - Introducción, en la que se presentará a la empresa ET.
 - Definición de objetivos y alcance del proyecto.
 - Consideraciones técnicas.
 - Descripción de la situación actual.

- Descripción básica de la solución propuesta (incluir diagrama)
- Justificación de la solución.
- Plan general para el proyecto (definiendo las etapas principales)
- Organización del proyecto.
- Definición de las responsabilidades de las partes.
- Criterios de aceptación del producto en sus diferentes etapas y en su terminación.
- Control de cambios.
- Propuesta económica, términos, forma de pago y vigencia
- Garantía.
- Contactos.
- Apéndices cuando se justifiquen en los que se detallará información relevante para el proyecto.

2.4 Análisis detallado

En primera instancia se definieron los estándares de nomenclatura

- **Estándar de nomenclatura de documentos.**

Se seguirá el siguiente formato:

PFAXXXXNNN

Donde:

P: Identificador del proyecto.

F: Fase a la que pertenece el documento.

A: Análisis.

D: Diseño.

C: Construcción.

A: Actividad a la que pertenece el documento.

XXXX: Identificación del documento basándose en la siguiente tabla.

NNN: En caso de ser varios documentos del mismo tipo se seguirá una numeración que irá en orden cronológico.

Tipo de documento	Identificación
PLANTRABAJOANALISIS.MPP	PTAN
INTANALISISUSR.DOT	IANU
REQRSMH.DOT	REQR
INICIO.DOC	INIC
ENTRSITACT.DOT	ENAC
ENTRSEG.DOT	ENSE
PROBLEMÁTICA.DOT	PROB
REQUERIMIENTO.DOT	REQF
DOCACEPTA.DOT	DOAC
MINUTA.DOT	MINU
ACEPTADOC.DOT	ACDO
ACEPANALISIS.DOT	ACAN
RETROALIMENTACION.DOT	RTAL
EVALUACION.DOT	EVAL
PLANTRABAJODISEÑO.MPP	PTDI
HW.DOT	HW
SW.DOT	SW
LP.DOT	LP
DBMS.DOT	DBMS
PRUEBAS.DOT	PR01: Pruebas por modulo. PR02: Pruebas de interfaz. PR03: Pruebas totales. PR04: Pruebas de carga.
DEFSW.DOT	DESW
DEFHW.DOT	DEHW
DEFINF.DOT	DEIN
DOCESTRAN.DOT	ESTR

También fue necesario definir la nomenclatura para los diagramas de análisis y diseño con la idea de que en toda la empresa se entienda lo mismo en cada dibujo.

Nomenclatura

Modelo relacional de objetos

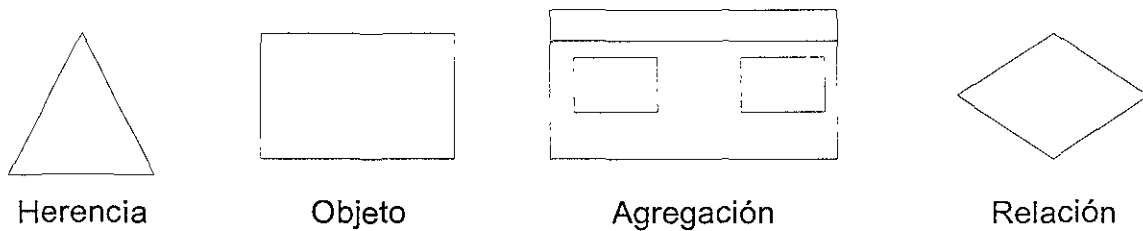


Figura 9 Modelo relacional de objetos

Donde:

Herencia:

Es el traspaso de propiedades y programas de unas clases a otras a través de la organización jerárquica a la que pertenecen. Esta propiedad facilita la reutilización de desarrollos anteriores en aplicaciones completamente nuevas, lo que las simplifica y reduce el esfuerzo necesario para construirlas. La herencia es unidireccional, pues una clase puede heredar de sus superclases, pero nunca de sus subclases. Existen dos tipos fundamentales de herencias:

- *Herencia simple.*- que se aplica cuando una clase puede ser generalizada por una sola superclase. (una superclase en cambio puede tener muchas subclases.)
- *Herencia múltiple.*- que se aplica cuando una clase puede ser generalizada por dos o más superclases independientes.

Objeto:

Es un conjunto de datos y programas que posee estructura y forma parte de una organización, es decir, no es un dato simple sino que contiene en su interior cierto número de componentes estructuradas y forma parte de una organización, jerárquica o de otro tipo.

Agregación:

Es la definición de nuevos objetos en términos de objetos construidos con anterioridad. Un objeto es parte o componente de otro.

Relación:

Es la forma en que interactúan diferentes objetos, hay tres tipos fundamentales de relaciones:

- Relaciones jerárquicas, de herencia o de generalización: especifican que una clase de objetos es subclase de otra más general. El conjunto de todas las relaciones de este tipo construye una o más jerarquías de clases. Pueden describirse con la frase *es un*. Por ejemplo: *Un diccionario consecutivo es un diccionario*.
- Relaciones de pertenencia o de agregación: indican que un objeto es parte o componente de otro, que a su vez es un agregado que contiene en su interior uno o más objetos de clase igual o diferente a la suya. Pueden describirse con la frase *es parte de*. Por ejemplo: *un par clave-valor es parte de un diccionario*.
- Relaciones de asociación: especifican cualquier otra relación entre dos objetos.

Esquema para la definición de métodos

Método Nombre objeto – método definido

Descripción :	Se explica en forma textual y concreta la finalidad del método.
Lectura :	Especifica la información que se consultará para realizar el proceso. En caso de que la palabra SUPPLIED anteceda al dato, señalará que es un dato que debe de ser proporcionado como parámetro.
Cambios :	Establece los cambios que se harán como resultado de la ejecución del proceso. Si antecede la palabra NEW a un campo o tabla, significa que se creará una nueva instancia del mismo.
Enviar :	Define la salida del proceso.
Precondición :	Se detallan las condiciones necesarias que se deben de cumplir para la realización del proceso.
Resultado :	Se determinan los pasos que se realizarán para llevar a cabo el proceso deseado.

Los lineamientos generales en la producción de un análisis se ilustran en la siguiente figura:

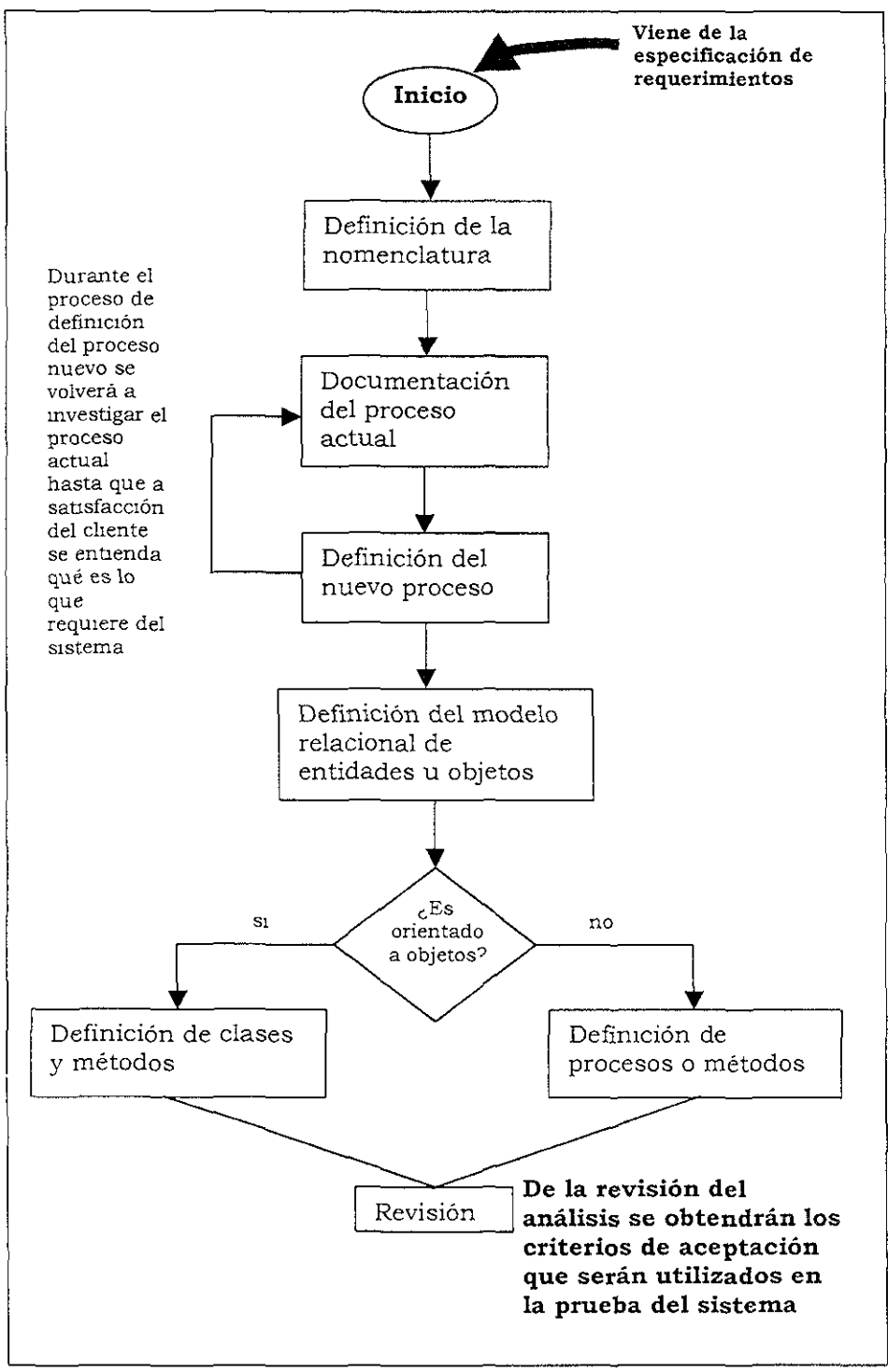


Figura 10 Diagrama de flujo para la realización de un análisis

Definición de la nomenclatura

En esta actividad se definen las convenciones que se usarán en el proyecto con la idea de que todos los involucrados realicen sus actividades en los mismos términos, en particular se debe definir:

- Nombres y claves para los documentos.
- Símbolos y sus significados para los diagramas.
- Objetos y métodos que deben contener:
 - Descripción.
 - Lectura.
 - Cambios.
 - Envíos.
 - Precondición.
 - Resultados.

Documentación del proceso actual

El analista junto con el cliente documentarán el proceso actual y verificarán que todas las fuentes de información necesarias para transformar las entradas del sistema actual en las salidas estén accesibles para el proceso.

Definición del nuevo proceso

Juntos el analista y el cliente plasmarán con todo el detalle la nueva forma en que habrá de realizarse el proceso, incluyendo el volumen esperado de transacciones y verificando que la información necesaria para transformar las entradas en salidas del sistema está accesible y en los formatos adecuados.

Se justificarán los cambios que habrán de realizarse y se listarán las restricciones del caso.

Definición del modelo relacional de entidades u objetivos

Las entidades u objetos son los componentes que tendrá la base de datos y sus relaciones entre ellas deberán quedar especificadas haciendo uso de un diagrama. La información que esta etapa genere será utilizada en el diseño de la base de datos en la etapa de diseño.

Definición de clases y métodos o definición de procesos o métodos (dependiendo de la filosofía con la que habrá de trabajarse en el proyecto)

En esta etapa el analista describirá cómo se habrán de transformar las entradas del sistema en salidas, es decir los procesos o métodos y en el caso de que se decida desarrollar el sistema orientándolo a objetos las clases y métodos que habrán de utilizarse.

Revisión del análisis

El encargado del proyecto junto con el cliente verificarán que en el análisis se presenten claramente los objetivos del proyecto, el alcance y que los procesos satisfagan las expectativas del cliente, es decir, que el análisis está bien hecho. Es conveniente que una vez terminado el análisis y entregado sea firmado por el responsable del proyecto por ET y por el cliente.

Resulta conveniente redactar un análisis utilizando un formato que facilite la interpretación, este formato es llamado por algunos autores *español estructurado* mismo que está descrito en el apéndice elementos de la administración de proyectos informáticos y aplica cuando la filosofía de programación es estructurada. Cuando está orientada a objetos ...

2.5 Diseño

En el siguiente diagrama se presenta el proceso para el diseño de un sistema.

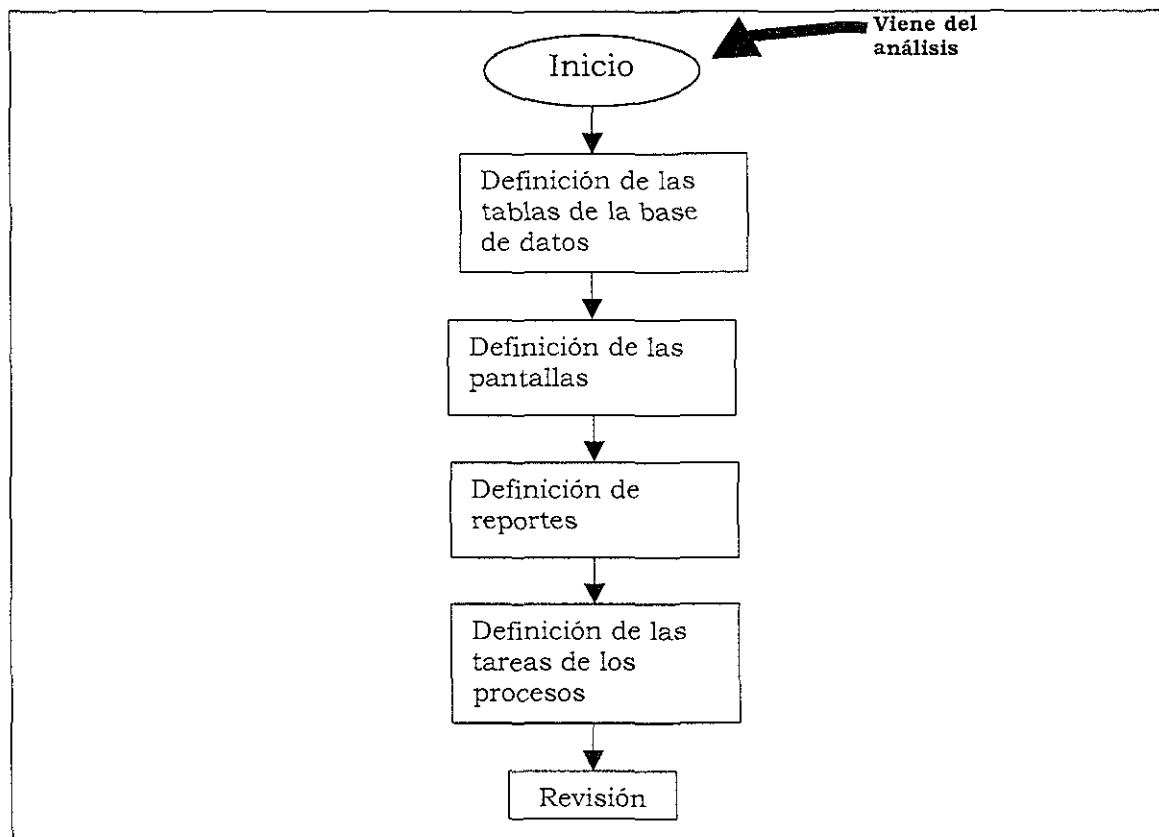


Figura 11 Diagrama de flujo para la realización de un diseño

Definición de las tablas de la base de datos

En las que se debe especificar:

- Definición.
- Nombre del analista que la diseñó.
- Fecha de diseño.
- Longitud por registro.
- El contenido de la tabla:
 - Campos.
 - Descripción
 - Tipo de datos.
 - Longitud.
 - Valor.

- Llaves (Guías para su localización)
- Relaciones entre las tablas

Definición de las pantallas

- Forma (imagen).
- Contenido (información que contendrán).
- Campos.
- Ligas con otras páginas.
- Menús que podrán desplegarse.
- Botones (tamaño, forma y localización)

Definición de reportes

- Encabezados
- Información que presentarán (contenido).
- Imagen.

Definición de tareas de los procesos

- Forma en que interactúan.
- Orden (jerarquía).

Algunos temas básicos que es conveniente tomar en cuenta sobre el diseño son:

- Definición de criterios para adoptar un estilo determinado para el diseño de cada proyecto (orientado a objetos, estructurado, u orientado a la estructura de datos).
- Creación de convenciones (estándares) dentro de la empresa para definir la modularidad, algoritmos básicos y estructuras de datos entre otros.
- Estipulación de criterios para diseñar los fundamentos de proyectos tipo (aplicaciones financieras, sistemas de tiempo real, o software de seguridad crítica entre otros.)

Deberá contemplarse la discusión de estos puntos antes de iniciar el proceso de diseño.

2.6 Estimación

Las estimaciones formales se realizan con el mismo procedimiento que se describe en la etapa de preanálisis con la diferencia de que en este nivel de avance del proyecto ya se cuenta con la información a detalle de lo que habrá de realizarse y no sólo una idea general.

2.6-a Tiempo

2.6-b Recursos humanos

2.6-c Recursos tecnológicos

Los recursos tecnológicos serán asignados con base en las necesidades del proyecto, ya sea que Enlaces Tecnológicos los asigne o compre o que el cliente los proporcione según las particularidades del caso.

2.7 Definición de indicadores de avance

Como complemento del plan de trabajo se debe hacer una lista de *los entregables* que cada tarea o actividad generará que servirán de ventana al comportamiento del plan del proyecto, mientras mayor sea la visibilidad, más fácil será detectar desviaciones de lo esperado. Estos entregables podrán ser de diversos tipos, documentos, submódulos del sistema, o especificación de algoritmos entre otros y se definirán en función de las condiciones particulares del proyecto o las necesidades del cliente. En general la identificación de los entregables ayuda a los responsables del proyecto junto con los clientes a establecer prioridades para las actividades.

Con los indicadores de avance se pretende dar al proyecto visibilidad en el progreso de desarrollo, un proyecto ideal tiene el 100% de visibilidad en su ciclo de construcción. En la siguiente imagen se muestra la visibilidad en un proyecto común, en un proyecto eficiente y en uno ideal.

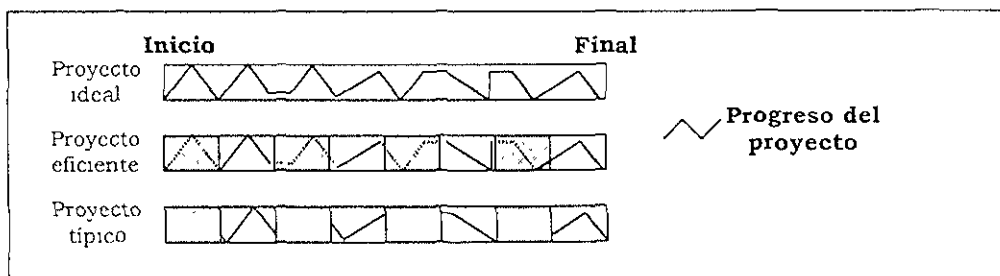


Figura 12 Visibilidad del avance en los proyectos informáticos

2.8 Definición de la forma de comunicación entre integrantes del equipo y con el cliente

- Foros de discusión (Intranet)
- Quiénes estarán en contacto con el cliente y cómo se llevará esta comunicación.
- Registro de los comunicados,...

2.9 Gestión de riesgos

Se puede iniciar el proceso utilizando como base la técnica desarrollada por Kepner y Tregoe *Análisis de Problemas Potenciales* para lo cual será necesario contestar las siguientes preguntas relativas a la situación particular del proyecto:

- ¿Qué podría salir mal?
- En caso de ocurrir ¿Cómo nos afecta?
- ¿Cuál es la probabilidad de ocurrencia o una *estimación* de la misma?
- ¿Cuáles son los puntos en los que nos afectaría más?
- ¿Qué podemos hacer al respecto ahora?
- En caso de que se presenten ¿qué acciones minimizarían su impacto?

El proceso para la gestión de riesgos es:

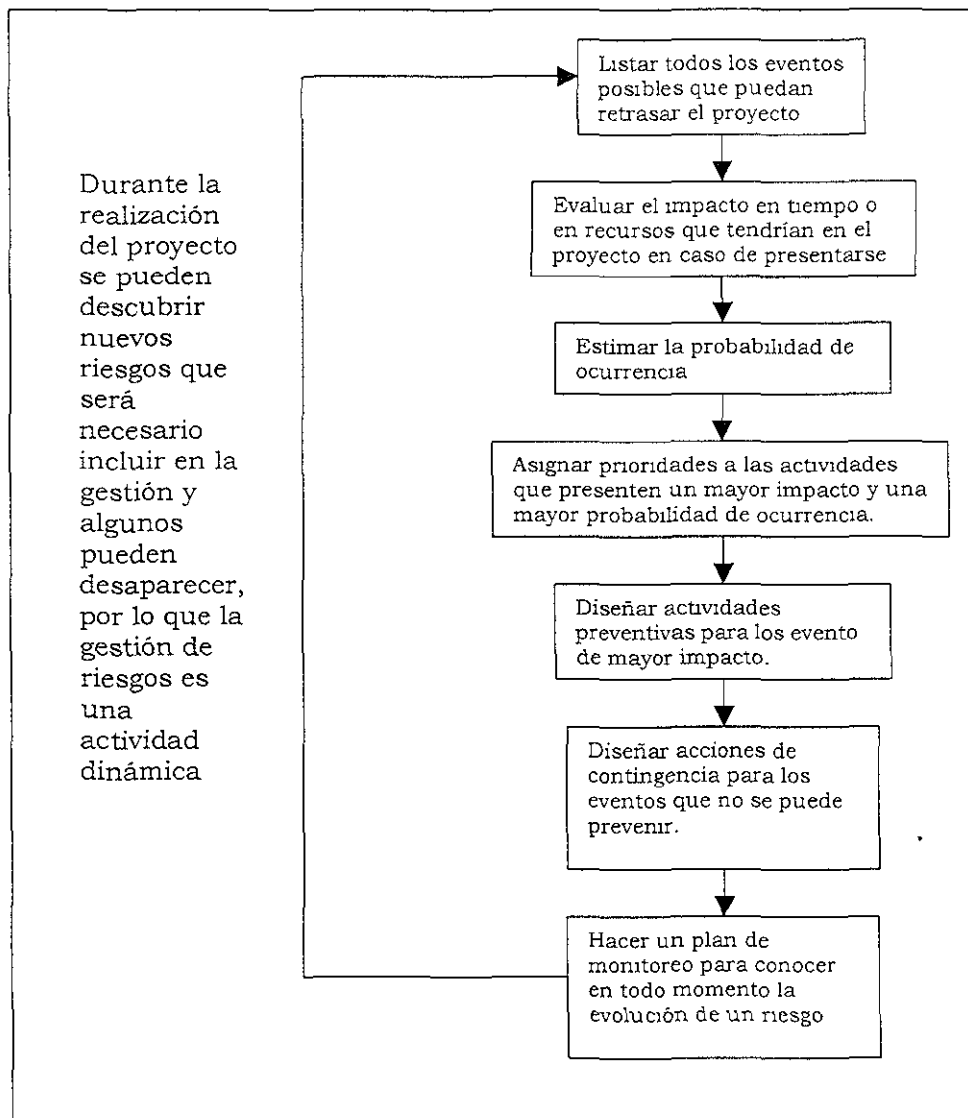


Figura 13 Diagrama de flujo para la gestión de riesgos

Según la experiencia de la gente de Enlaces Tecnológicos se puede identificar dos eventos que a lo largo de su experiencia en la creación de sistemas son una constante y que retrasan los proyectos por lo que vale la pena describirlos para, primero hacer un esfuerzo importante por evitar que ocurran, dar seguimiento a su gestación en todo momento y finalmente proponer acciones de contingencia que minimicen su impacto. Estos eventos son:

- La falta de información que se da a los analistas y el poco interés que el cliente brinda a conseguirla.
- Que para hacer las pruebas del sistema no se cuente con el equipo y ambiente adecuado.

2.10 Plan del proyecto

La forma en que se organizan las actividades en un proyecto definido se ilustra en le siguiente diagrama.

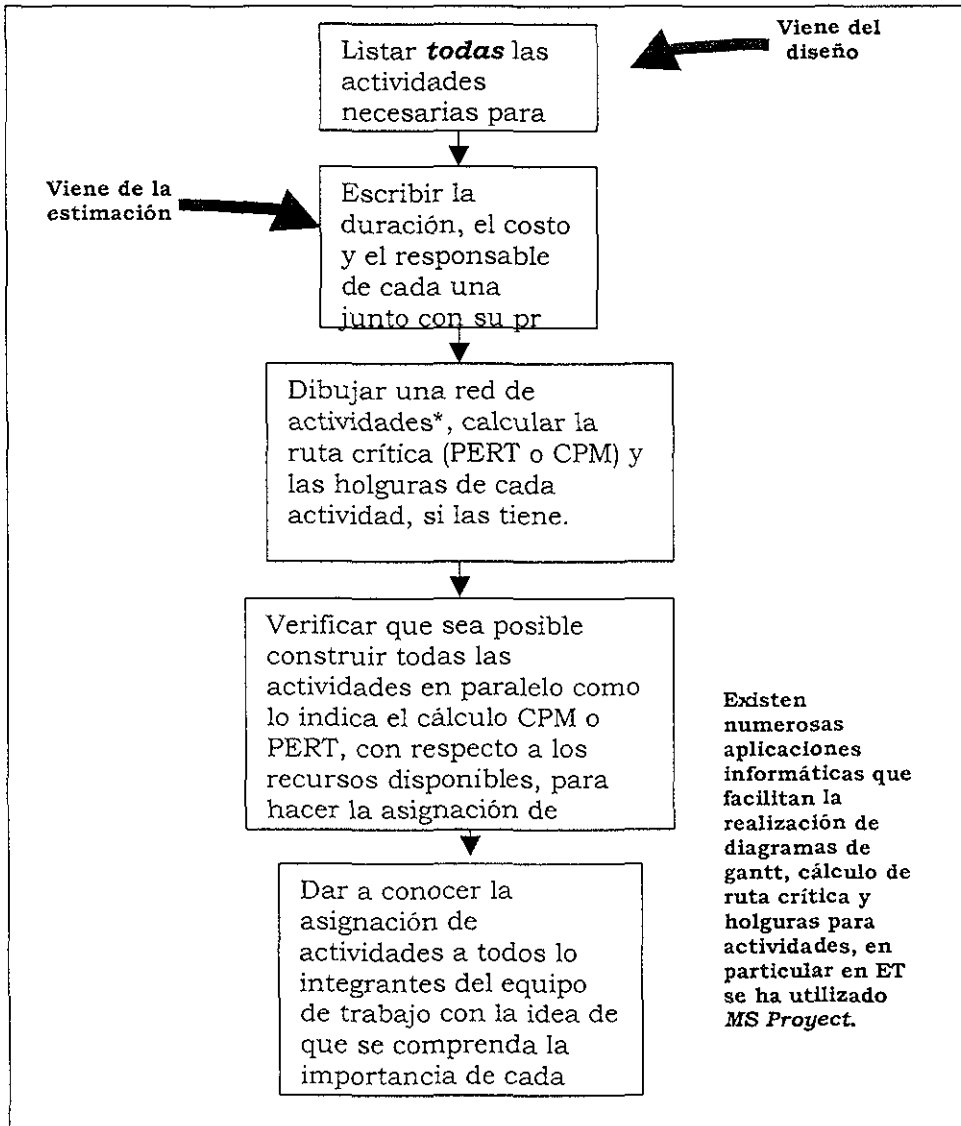


Figura 14 Diagrama de flujo para realizar un plan de trabajo

Es común que los proyectos sean expresados en términos de conjuntos de tareas cuando éstas son en extremo numerosas, la habilidad del responsable del proyecto en agruparlas y asignar responsables junto con la gestión de riesgos aumentarán la probabilidad de llevar a buen término el proyecto.

Es importante mencionar que dentro de las actividades del proyecto suele pasar por alto la documentación del sistema, y esto es un error que ocasiona atrasos importantes (por ejemplo hacen difíciles las pruebas porque no queda claro que se esperará del sistema, o no se podrá monitorear su desempeño) y una baja satisfacción de los clientes.

2.11 Construcción

Dentro de la empresa Enlaces Tecnológicos se ha adoptado el uso de una herramienta para la construcción de sistemas en la que intervienen muchas personas que facilita la visibilidad en la construcción y con ello su seguimiento por parte del responsable. Esta herramienta se llama CVS (Control Version System) y en el siguiente diagrama se explica su uso.

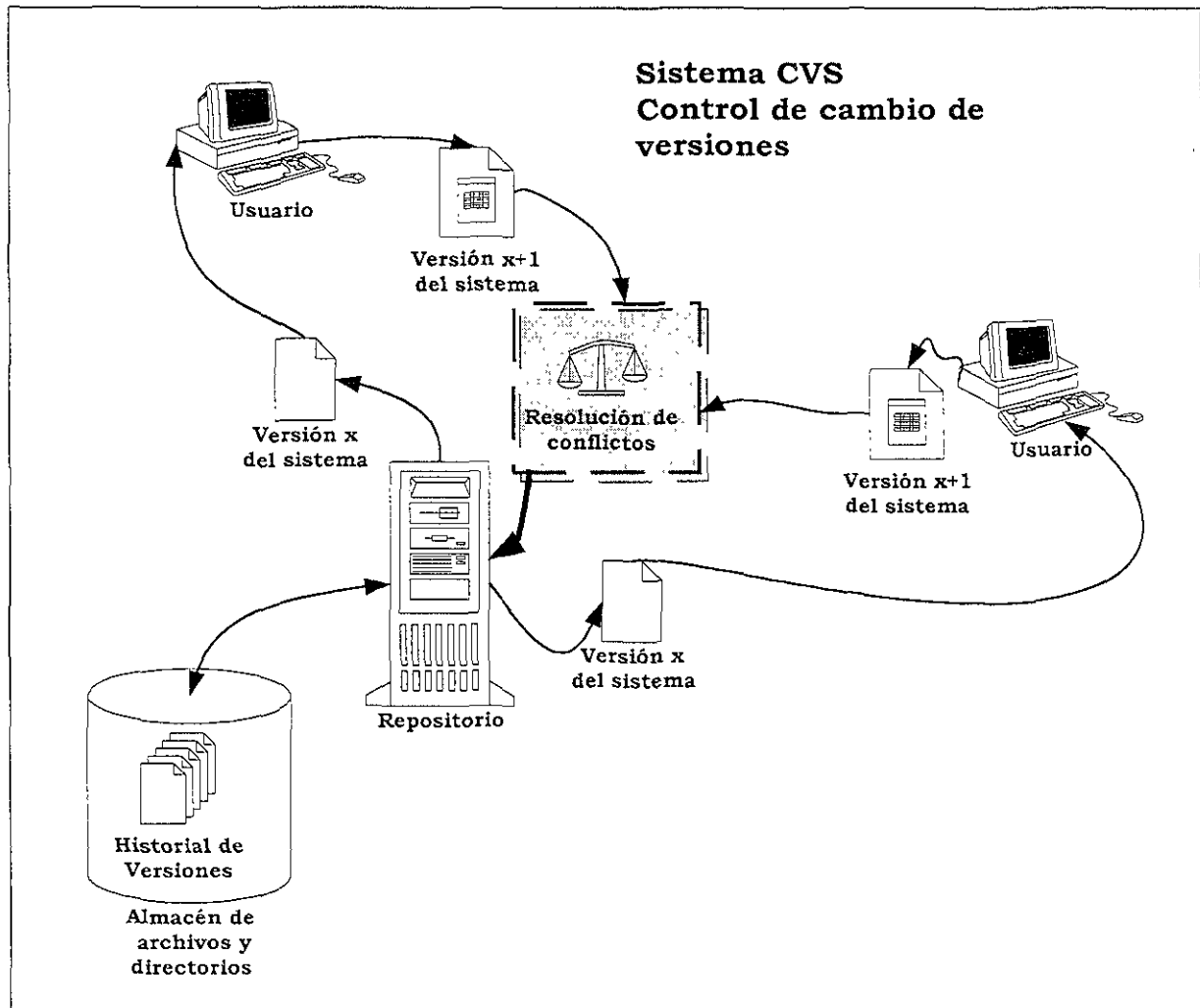


Figura 15 Diagrama que muestra el funcionamiento del CVS

Las versiones del sistema se almacenan en el repositorio, cada usuario puede obtener la última versión del sistema para trabajar con ella, una vez que lo ha hecho devuelve el archivo modificado que será la nueva versión. Por otro lado otra persona pudo haber utilizado la última versión para modificarla, en caso de que así sea, cuando mande su versión modificada el sistema CVS verificará que no existan conflictos

entre las diferentes modificaciones, si existen le notificará a la persona que quiere hacer una actualización de la versión que existen conflictos y el sistema se los mostrará línea por línea para que vea cuál es la discrepancia.

Por otro lado, cada que se hace una actualización de la versión el sistema le pide al que la hizo que describa los cambios que efectuó y los almacena de forma que siempre que se consulte la historia de un archivo se presentarán todas las versiones que se han hecho con una descripción de los cambios entre cada una. En el apéndice de elementos de la administración de proyectos se describe mejor el sistema CVS.

También es importante definir convenciones que faciliten la lectura de programas escritos por diferentes miembros de la empresa; algunos parámetros que se pueden estandarizar son:

- Métodos de codificación (incluyen las variables, las funciones, presentación y nombramiento de documentos.)
- Conceptos relativos a los datos (incluyendo el ámbito, persistencia y momento de asignación.)
- Directrices para utilizar tipos de datos específicos (incluyendo los números en general, enteros, números con punto flotante, caracteres, cadenas de caracteres, booleanos, tipos enumerados, constantes, vectores y punteros.)
- Conceptos relativos al control (tomando en cuenta la organización lineal del código, uso de condicionales, control de loops, utilización de expresiones booleanas, control de la complejidad y organización de estructuras de control poco usuales, como *goto*, *return* y procedimientos recursivos.)
- Reglas para compactar código dentro de rutinas, módulos, clases y archivos.
- Métodos de depuración y comprobación de unidades.
- Estrategias de integración (como la integración incremental, integración *big-bang* y desarrollo evolutivo.
- Estrategias y métodos para el afinamiento del código.
- Estipulación de lineamientos para escoger el lenguaje de programación adecuado.
- Uso de herramientas de construcción (como entornos de programación, soporte para grupos de trabajo como el correo electrónico, y el control de código fuente, bibliotecas de código y generadores de código.)

2.12 Control

Se deben definir criterios para dar seguimiento a los proyectos de forma que convenga a los intereses de los mismos y satisfaga las preferencias de Enlaces Tecnológicos, algunos modelos para controlar el avance de los proyectos se enlistan a continuación y se explican en el apéndice I elementos de la administración de proyectos informáticos.:

- Modelo de cascada pura
- Espiral.-
- Prototipado evolutivo.-
- Entrega por etapas.-
- Diseño por planificación.-
- Diseño por herramientas.

Estará en manos del líder del proyecto decidir cuál o cuáles estrategias para fijar los puntos de control se usarán a lo largo del proyecto. Muchas veces es el cliente el que toma esta decisión.

2.13 Pruebas de calidad

Las pruebas de calidad de un sistema son de tres tipos, de funcionalidad (modularidad), de integración y de volumen o de estrés.

- Las pruebas de funcionalidad deben verificar que cada módulo se comporte tal y como dice la especificación.
- Las pruebas de integración deben constatar que el acoplamiento de los módulos funcione correctamente, es decir, que la aplicación trabaje de forma adecuada al hacerlo como un todo. Y no afecte el sistema o los demás sistemas con los que debe convivir.
- Las pruebas de volumen o de estrés comprobarán que el sistema se comporte adecuadamente para condiciones de carga o de volumen de transacciones extremas.

Las pruebas de calidad deben ser planeadas y en este sentido debe hacerse una matriz que describa lo que ha de probarse, los resultados que se deben obtener y los criterios de aceptación que han sido especificados con anterioridad en la etapa de análisis del sistema. Como se mencionó en el apartado de la gestión de riesgos es muy común que para probar un sistema sea necesario crear un ambiente de pruebas y a

veces hasta un software especial que genere las condiciones extremas de carga.

Para seguir los lineamientos de la Norma ISO 9002 sobre los sistema de información se debe tomar en cuenta los siguientes aspectos básicos:

▪ **Aseguramiento del software:**

- Documentados los requerimientos y actividades del plan.
- La fecha de implementación de las actividades del plan y cuál será su presupuesto
- La interacción entre el sistema de aseguramiento y los esfuerzos de construcción.
- La participación del plan de aseguramiento en todo el proceso de pruebas.

▪ **Desarrollo de la documentación**

- Estándares para la preparación de la documentación.
- Documentación de los estándares.
- Procedimientos establecidos y documentados que aseguren el seguimiento de la estandarización.
- Procedimientos para manejar los cambios en la documentación.
- Contenidos de la documentación claros.
- Procedimientos para mantener la consistencia en la escritura.
- Definición de la autoridad que aprobará la documentación.
- Procedimientos establecidos para la producción de documentación impresa.

▪ **Código**

- Estándares y convenciones definidas.
- Completos de pseudocódigo descripción de elementos de los datos y subrutinas.
- Consistencia entre código y pseudocódigo.
- Identificación del código depurado.
- Identificación de correcciones.

▪ **Administración de la configuración**

- Plan de configuración
- Lineamientos del plan.
- Disponibilidad para consultar el plan para los participantes.
- Procedimientos de documentación de problemas de configuración.
- Publicación de agenda y minutas.

▪ **Librerías de programas**

- Procedimientos para controlar las librerías.
- Disponibilidad para revisión
- Control de actualización de librerías.
- Registro de ubicación con número de librería y versión.

▪ **Reportes no conformados y Acciones correctivas**

- Procedimientos para detectar deficiencias.
- Estudios de tendencia de desempeño. Documentación de acciones correctivas.
- Grupo de soporte para acciones correctivas.

▪ **Verificación y Validación**

- Determinación de las pruebas con base en el análisis de requerimientos.
- Objetivos de pruebas que garanticen su fiabilidad.
- Las pruebas deben ser entendidos por el personal del proyecto
- Pruebas funcionales, integrales y de volumen según se acordó en los criterios de validación.

▪ **Estatus del proyecto**

- Comparación de lo planeado con lo real.
- Análisis de la discrepancia.

- Análisis de causas de discrepancia y comparación con los planes de contingencia y las actividades de control.

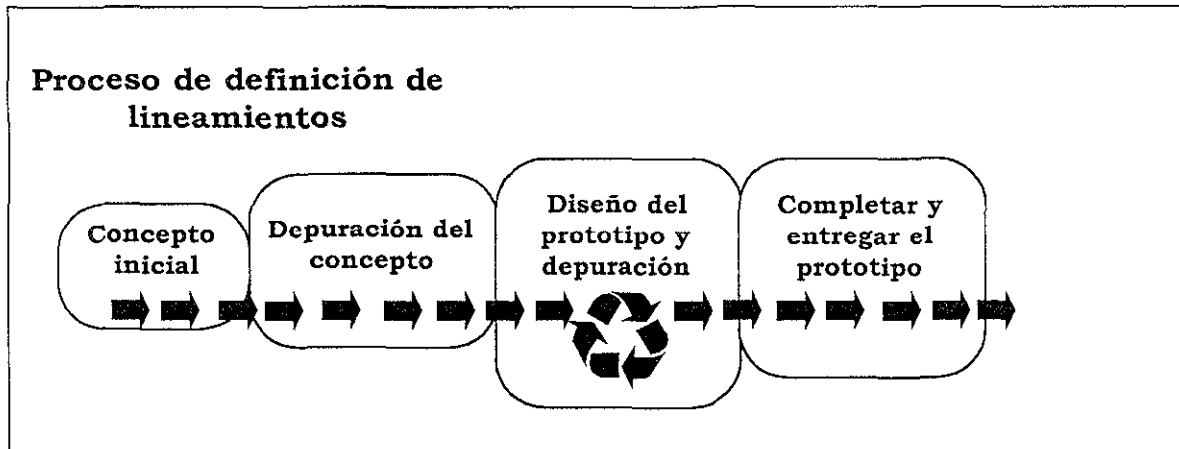
2.14 Entrega e instalación

Se acordó que dentro de la planeación del proyecto se debe tomar en cuenta todos los factores que puedan impedir el uso cotidiano del sistema por parte del cliente, o sea, que además de actividades como la documentación técnica, y la elaboración de manuales para los usuarios, se estudiará con cuidado la forma en que habrá de ponerse a punto el sistema, con la idea clara de que el sistema debe servir, en primera instancia, para que el cliente lo use. Así que el responsable del proyecto por parte de ET deberá supervisar junto con el cliente la instalación del sistema y ayudarlo en la medida de lo posible, a veces como se menciona en el apéndice de elementos de la administración de proyectos informáticos la instalación y puesta a punto de un sistema puede resultar en sí un proyecto complejo.

2.15 Plan de seguimiento posventa

Finalmente se le asignará la responsabilidad a alguna persona de ET de verificar que el sistema se comporte con un desempeño adecuado al paso del tiempo, por una parte debe preocuparse por la entera satisfacción del cliente y por otra fomentar una relación de largo plazo con el mismo haciendo que el conocimiento que ET ha adquirido de su negocio le permita participar en nuevos proyectos.

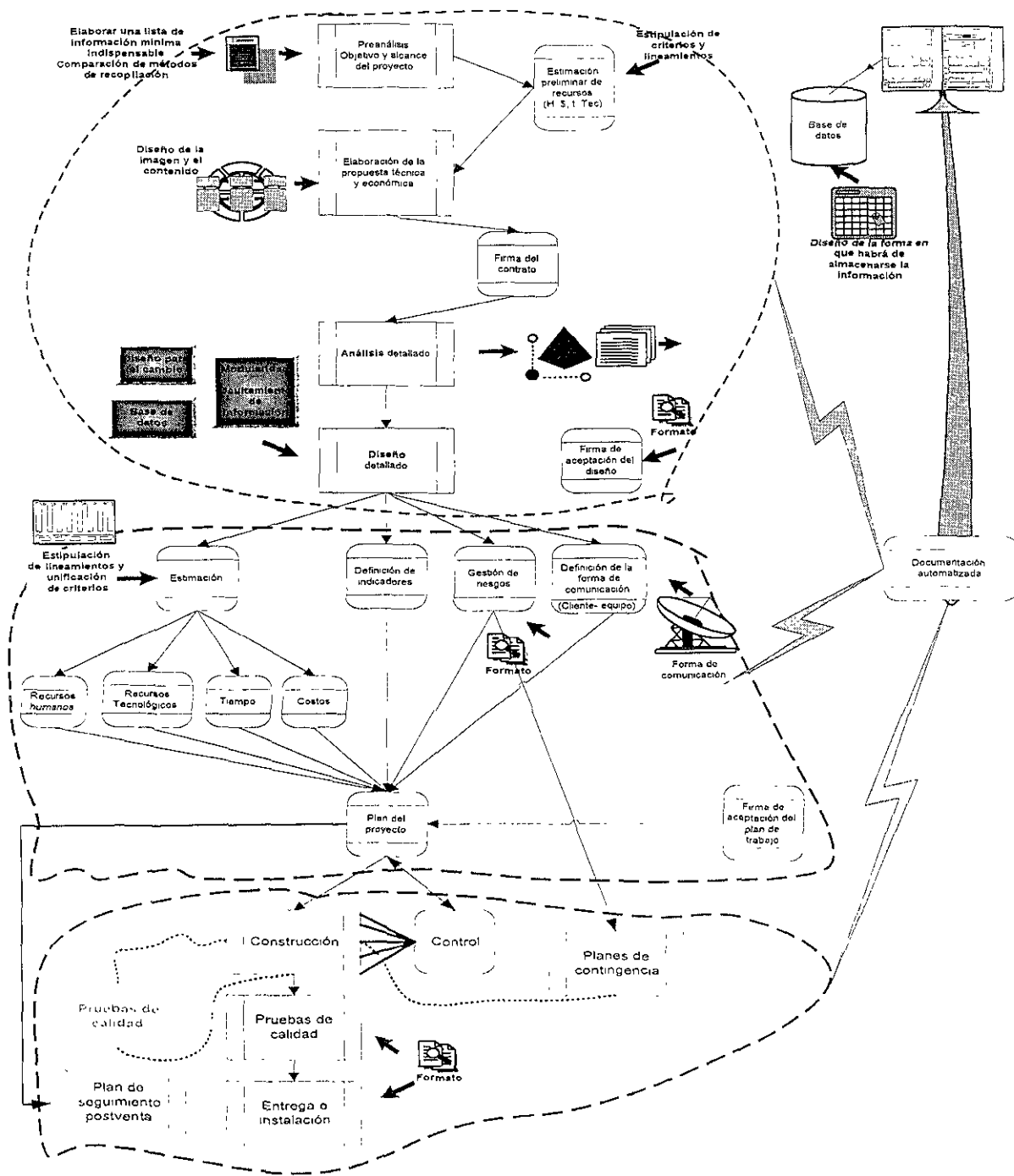
2.16 Dinámica para la especificación



Tras haber elaborado, sintetizado y complementado la información de las encuestas que se realizaron en la empresa y con algunas personas expertas en el medio informático, se realizó un diseño preliminar de cada elemento a definir en el procedimiento, se presentó a algunos participantes y se mejoró, para realizar con esto el primer prototipo que sería sometido a consideración de la dirección de la empresa, para revisarlo y corregirlo hasta que fuera aprobado.

Esta actividad dejó el precedente de la mejora continua en el diseño de actividades y procedimientos porque resultó evidente que a pesar de tener un lineamiento para la empresa definido y aceptado por todos se mejoró en poco tiempo.

2.17 Digrama de flujo del procedimiento para la administración de proyectos informáticos en Enlaces Tecnológicos S.A de C.V



3. Conclusiones

Como conclusión de este trabajo se listarán a continuación tres aspectos que resultan importantes de señalar:

Situación inicial Vs. situación final

- Como resultado de la interacción del nuevo procedimiento de administración de proyectos con la forma en que se llevaba la administración propia de la empresa fue necesario diseñar una estructura de costos que permitió unificar los criterios para hacer propuestas y cotizaciones en un menor tiempo. En particular estos puntos son la aportación de mi trabajo para la administración de proyectos dentro de la empresa:
 - Se definió un algoritmo empírico (basado en la experiencia de los participantes de la empresa y con la idea de seguir adecuándolo conforme se presenten nuevas experiencias) para hacer las estimaciones futuras dentro de la empresa con base
 - Se especificó el margen de utilidad base de la empresa dentro de un proyecto y los parámetros que deben tomarse en cuenta para modificarlo.
- Hacer del conocimiento de todos los participantes en la empresa los conceptos de gestión de riesgos ha permitido reducir el impacto de situaciones adversas en proyectos que se llevan en la actualidad, aún en los que no se utiliza el procedimiento definido en esta tesis al no tener ET a su cargo la administración del proyecto.
- La definición de un esquema para la asignación de recursos humanos en los proyectos llevó a identificar la necesidad de tener una estructura administrativa que ayude a evaluar el desempeño del personal en aspectos tanto técnicos como de relaciones humanas, personalidad, actitud y compromiso hacia la empresa.
- El procedimiento en sí ha ayudado a los gerentes de proyectos en ordenar de forma sistémica la forma de conceptuar todo el proyecto y sus implicaciones desde el principio, lo que evita la omisión de actividades necesarias (como la documentación o la instalación y puesta a punto del producto) en el plan original, lo que resultaría en atrasos, costos y descontento de los clientes.
- También se tuvieron avances por la especificación de la forma y contenido de las propuestas que hace la empresa.

Condiciones para que funcione un procedimiento nuevo

- Para que el procedimiento de administración de proyectos así como cualquier forma nueva de realizar las actividades cotidianas sea implantada y seguida dentro de cualquier empresa deben cumplirse ciertos requerimientos, entre otros:
 - Los objetivos del cambio deben ser claros para todos.
 - El cambio debe contener ideas utilizables de inmediato en la práctica.
 - Se debe poder modificar el procedimiento en función de su uso y el grado de dominio que se tenga en la empresa de sus implicaciones en las situaciones cambiantes de su entorno.
 - La dirección debe estar obligada a seguir el nuevo procedimiento antes que pueda exigir que sea utilizado en todos los niveles de la compañía.
 - Se le debe dar seguimiento a su desempeño continuamente.

Continuación del trabajo

- Se aprobó una partida presupuestal para automatizar el procedimiento dentro de una Intranet que tiene la empresa, siguiendo el procedimiento definido en esta tesis para su administración.
- Se piensa continuar la mejora del procedimiento con algoritmos de verificación de consistencia para la estimación de recursos (humanos, de dinero, equipo y tiempo) para futuros proyectos.
- Se definirá un modelo para optimizar las utilidades basándose en los algoritmos de costos y tiempos de un proyecto que se plantearon en este trabajo con la intención de optimizarlo.

Para finalizar se debe decir que los objetivos, táctico-operativos, estratégicos y normativos que se plantearon al comienzo de este trabajo se cumplieron de forma satisfactoria.

Apéndice I

• Elementos de la administración de proyectos informáticos

La administración de proyectos, de cualquier tipo, consiste en planear y controlar un proyecto; Su finalidad es garantizar que se logren los objetivos del proyecto en el tiempo previsto y con el presupuesto asignado.

Resulta pertinente hacer también una definición de lo que se debe entender por **proyecto**. Proyecto es un conjunto de actividades finitas que, por lo general, se realizan una sola vez (en conjunto) y que tiene objetivos claramente definidos.

El tiempo que transcurre entre que un proyecto comienza y termina es conocido como el ciclo de vida del proyecto, algunos autores dividen este tiempo en cuatro fases en las que se realizan actividades de diferente índole, en la siguiente tabla se muestran junto con algunas actividades típicas en esas fases; y en seguida se explicarán con más detalle.

Fase	Actividades típicas
De concepción	<ul style="list-style-type: none">▪ Preanálisis▪ Diseño preliminar (a grandes rasgos)▪ Evaluación económica
Formativa	<ul style="list-style-type: none">▪ Análisis formal▪ Diseño detallado▪ Estimación de recursos y esfuerzo▪ Definición de puntos de control▪ Gestión de riesgos▪ Estipulación del sistema de comunicación▪ Plan de trabajo
Operativa	<ul style="list-style-type: none">▪ Construcción▪ Control (acciones de contingencia, si aplican)▪ Pruebas de calidad
Terminación	<ul style="list-style-type: none">▪ Entrega e instalación▪ Evaluación de resultados▪ Seguimiento del desempeño▪ Mantenimiento (si aplica)

Fases y actividades típicas de la administración de proyectos informáticos

La siguiente tabla muestra los porcentajes de tiempo sugeridos en proyectos *tipo* divididos en dos categorías, obtenidos de el artículo de Steve McConnell (1993) *code complete*.

Actividad	Proyecto pequeño (menos de 2500 líneas de código)	Proyecto grande (mas de 2500 líneas de código)
Análisis	10%	30%
Diseño detallado	20%	20%
Codificación / depuración	25%	10%
Prueba de unidades	15%	20%
Integración	15%	20%
Prueba del sistema	10%	15%

- **Fase de concepción.-** En la que se estudia el beneficio y la viabilidad de llevar a cabo el proyecto para decidir si se realiza o no.

Para las empresas que desarrollan software a la medida de las necesidades de clientes en esta fase se encuentran: Los aspectos que hacen que el cliente potencial detecte una necesidad informática o decida que es conveniente llevar a cabo algún esfuerzo en este sentido, seguidamente se buscará la forma de evaluar el costo de tal acción ya sea encargándosela a personal interno o contratando externos, cualquiera de los caminos que sea seguido continua con la definición de requisitos para definir explícitamente los objetivos generales y el alcance del proyecto, para realizar un análisis preliminar que permita tener una idea de la solución, el plan de trabajo, el equipo, personal, tiempo y dinero necesarios para llevarlo a cabo y así evaluar la conveniencia del proyecto con los criterios que sean pertinentes.

Principales elementos y estrategias de un preanálisis:

Definición del preanálisis: es la etapa en la que se definen básicamente los objetivos del proyecto y su alcance.

Lo que se busca hacer es y algunas formas de lograrlos son:

Una especificación mínima:

Que es la cantidad mínima de información necesaria para especificar de forma comprensible el producto. Se puede hacer de cualquiera de las siguientes formas:

- *Una especificación resumida.* Es una descripción del software que desea construir en forma de texto, que debe tener un máximo de diez cuartillas.
- *Especificación del partida.* Es una especificación aproximada inicial, no realizada pensando en mantenerla una vez escrita. Su objetivo primario es dar una visión común del producto al grupo de desarrollo, clientes y usuarios finales. Una vez esta unificación, la especificación ha cumplido su propósito, y no necesita ser mantenida.
- *Manual de usuario como especificación.* En lugar de escribir una especificación tradicional, se debe escribir un manual para el usuario, y se hace que el software se adapte al manual. Como de todos modos se debe escribir dicho manual en algún momento, podría escribirse primero y eliminar la redundancia en que incurriría escribiendo el manual de usuario y una especificación. Una variante de esta idea consiste en escribir un sistema de ayuda en línea para su uso como especificación.

- *Prototipos de interfaz de usuario.* Puede servir como una especificación en sí, o puede complementar una especificación escrita. Si se crea cuidadosamente, una imagen puede valer más que mil palabras, y puede necesitar mucho menos tiempo para su creación.
- *Cuadernos.* En ocasiones, la tecnología más simple es la que funciona mejor. Si se trabaja con clientes o usuarios finales que pueden visualizar el software sin tener que verlo en la pantalla de una computadora, a veces resulta más rápido y fácil dibujar informes, pantallas y otros elementos de la interfaz de usuario sobre el papel, y diseñar el producto de esa forma.
- *Definición del alcance.* Cree una definición del alcance que describa lo que hay que incluir y lo que hay que dejar fuera del producto. Describir lo que va a quedar fuera del producto resulta difícil, pero es la función esencial de este tipo de definición del alcance. La tendencia natural consiste en hacer un producto que sirva para todo y para todo el mundo, y una buena definición del alcance marca claramente lo que *no* es un producto.
- *Objetivo global del producto.* Se debe crear un objetivo global del producto. Los objetivos globales están relacionadas con las visiones, y los mecanismos excelentes para evitar la *prestacionitis*.

Se pretende hacer una propuesta de solución en un intervalo de tiempo apropiado para la complejidad del proyecto y las expectativas del cliente;

Se debe elaborar una lista de la información **mínima** indispensable para poder hacer una propuesta.

Estipulación de criterios y lineamientos

Para hacer una propuesta de desarrollo de forma estándar es necesario definir criterios y lineamientos para que los encargados de esta actividad tengan bases para que, independientemente de quién la haga, el resultado sea muy parecido, los aspectos a definir son:

- Definición del esfuerzo necesario para realizar el proyecto.
- Personal que será utilizado, a quién(es) se le(s) encomendará el trabajo y cómo habrán de organizarse.
- Tiempo que será necesario para terminar el proyecto y cómo debe hacerse para estimarlo.
- Tecnología que deberá utilizarse.
- Equipo que se utilizará.
- Costos para Enlaces.
- Criterios para definir el margen de utilidad sobre los costos de producción.

Formato de propuesta

La forma en que se presenta una propuesta da al cliente una idea de los que puede esperar de la compañía proveedora a cambio de los que se pagará por el servicio, así que es importante definir tanto la forma como el contenido de la misma.

- **Fase formativa (planeación).**- En la que se definen con claridad los objetivos, el tipo de organización y se nombra al administrador del proyecto. Luego, se realiza un **análisis detallado** y el **diseño** del programa o sistema que son los aspectos clave para elaborar el plan de trabajo maestro del proyecto y con este se elaborarán con detalle programas, requerimientos de recursos y presupuestos. La planeación del proyecto se realiza con la idea de prever los problemas y asegurarse que se cuente con los recursos necesarios en el momento oportuno, para lo que es conveniente que todos los interesados deben participar en esta etapa, y se define con la mayor claridad posible la forma de comunicación que habrá de tenerse entre los participantes del proyecto mientras éste dure. También en esta fase se concibe la forma en que habrá de darse seguimiento y control al trabajo para que, en la medida de lo posible, se estipulen las acciones posibles de contingencia cuando sea necesario.

Resulta pertinente comentar que en general dentro de las organizaciones, el análisis y diseño de sistemas se refiere al proceso de examinar la situación de una empresa con el propósito de mejorarla con métodos y procedimientos más adecuados; más en concreto: *el diseño de sistemas* es el proceso de planear, reemplazar o complementar un sistema organizacional existente. Pero antes de llevar a cabo esta planeación es necesario comprender, en su totalidad, el viejo sistema y determinar la mejor forma en que se pueden, si es posible, utilizar las computadoras para hacer la operación más eficiente. *El análisis de sistemas*, es por consiguiente, el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema, es decir, en el análisis del sistema se trata de comprender cómo opera el sistema para identificar con precisión los requerimientos del mismo.

Análisis detallado

Lo que se persigue en esta etapa es delimitar, de la forma más detallada posible, los procesos que habrán de ser automatizados, comprender el ambiente en el que interactúan y los componentes ajenos con los que lo harán y la forma en que habrá de comunicarse el usuario con el sistema. A partir de que se tiene esta información se debe tener la información necesaria para especificar detalladamente los requerimientos del cliente.

- Para el análisis se suelen utilizar herramientas de tres tipos: para recolección de datos, para realizar diagramas y para almacenar descripciones de los elementos del sistema. Básicamente se deben contestar las siguientes preguntas: ¿qué procesos integran el sistema?, ¿qué datos emplea cada proceso?, ¿qué datos serán almacenados? Y ¿qué datos entran y salen del sistema?

Resulta conveniente utilizar métodos de redacción orientados a evitar problemas de ambigüedad en el lenguaje, un de ellos es el español estructurado (cuando la filosofía de programación es estructurada) que no hace uso de árboles o tablas; en su lugar utiliza declaraciones para describir el proceso. El método no muestra las reglas de decisión; las *declara*.

Aún con esta característica, las especificaciones en español estructurado requieren que el analista primero identifique las condiciones que se presentan en un proceso y las decisiones que se deben tomar cuando esto sucede, junto con las acciones correspondientes. Sin embargo, este método también le permite hacer una lista de todos los pasos en el orden en que se llevan a cabo. Para ellos no se utilizan símbolos ni formatos especiales, características de los árboles y tablas decisión que para algunos resultan incómodos. Además, es posible describir con rapidez los procedimientos en su totalidad ya que para ello se emplean declaraciones muy similares al español.

La terminología utilizada en la descripción estructurada de una aplicación consiste, en gran medida, en nombres de datos para los elementos que están definidos en el diccionario de datos desarrollado para el proyecto.

Desarrollo de declaraciones estructuradas. El español estructurado emplea tres tipos básicos de declaraciones para describir un proceso; estructuras de secuencia, estructuras de decisión y estructuras de iteración. Estas estructuras son adecuadas para el análisis de decisión y pueden trasladarse al desarrollo de software y programación.

Estructuras de secuencia. Una *estructura de secuencia* es un solo paso o acción incluido en un proceso. Éste no depende de la existencia de ninguna condición y, cuando se encuentra, siempre se lleva a cabo. En general, se emplean varias instrucciones en secuencia para describir un proceso.

Estructuras de decisión. El español estructurado es otro camino para mostrar el análisis de decisión. Por tanto, a menudo se

incluyen las secuencias de acciones dentro de estructuras de decisión que sirven para identificar condiciones. Es así como las estructuras de decisión aparecen cuando se pueden emprender dos o más acciones, lo que depende del valor de una condición específica. Para esto, primero se evalúa la condición y después se toma la decisión de emprender la o las acciones asociado con esta condición. Una vez determinada la condición las acciones son incondicionales, como ya se ha mencionado.

A continuación se presenta un ejemplo relacionado con la compra de un libro para ilustrar las estructuras de decisión:

Si se encuentra el libro deseado entonces

Llevar el libro al mostrador de salida.

Pagar el libro.

Asegurarse de obtener el recibo de compra.

Abandonar la librería.

De otro modo

No llevar libros al mostrador de salida.

Abandonar la librería.

La estructura de decisión que emplea las frases *si / entonces / de otro modo*, señala con bastante claridad las alternativas del proceso de decisión. En este caso se indican dos condiciones y dos acciones. Nótese la forma en que cada secuencia está contenida dentro de cada condición. La parte *si* contiene cuatro declaraciones separadas en secuencia, mientras que la parte *de otro modo* contiene sólo dos, situación que es bastante común. Con el español estructurado se pueden incluir estructuras dentro de otras estructuras. Por otra parte, escribir los detalles en un formato que contenga sangrías es de gran ayuda para agrupar condiciones y acciones. La claridad adicional que se obtiene con esta convención tiene mayor utilidad cuando se consideran situaciones de decisión amplias o complicadas.

Estructuras de iteración. En las actividades rutinarias de operación, es común encontrar que algunas de ellas se repiten *mientras* existen ciertas condiciones o *hasta* que éstas se presentan. Las instrucciones de *iteración* permiten al analista describir estos casos.

La búsqueda de un libro en la librería puede realizarse repitiendo los siguientes pasos:

Ejecutar mientras se examinan más libros;

Leer el título del libro.

Si el título suena interesante

Entonces tomar el libro y hojearlo.
Buscar el precio.
Si la decisión es llevar el libro
 Colocarlo en la pila de *libros para llevar*.
 Otro regresar el libro al estante.
Fin de si
Otro continuar

Fin de ejecutar

Si se encuentran los libros deseados *entonces*

 Llevar los libros al mostrador de salida.

 Pagar los libros.

 Asegurarse de obtener el recibo.

 Abandonar la librería.

Otro

 No llevar libros al mostrador de salida.

 Abandonar la librería.

Fin de si.

En este ejemplo se observa que se describen cero, una o más iteraciones para describir la búsqueda de los libros de interés. Los pasos adicionales anidados dentro del lazo de iteración, dan instrucciones sobre qué hacer cuando el título parece interesante o cuando se halla el libro deseado. La repetición de este proceso continúa siempre y cuando exista la condición de “examinar más libros”. Después el procedimiento indica los pasos a seguir en caso de que se encuentren libros de interés.

Como se ha observado el español estructurado puede ser de utilidad para describir con claridad condiciones y acciones. Cuando se examina el ambiente de una empresa, los analistas pueden utilizar el español estructurado para declarar las reglas de decisión que se aplican en ese medio. Si los analistas no pueden declarar qué acción emprender cuando se toma una decisión entonces necesitan adquirir mayor información para describir la situación. Por otro lado, después de describir las actividades en forma estructurada, los analistas pueden pedir a otras personas que revisen la descripción y determinen con rapidez los errores u omisiones cometidos al establecer los procesos de decisión.

Diseño

Algunos temas básicos sobre la arquitectura y el diseño son:

- Definición de criterios para adoptar un estilo determinado para el diseño de cada proyecto (orientado a objetos, estructurado, orientado a la estructura de datos...)
- Creación de convenciones dentro de la empresa para definir la modularidad, algoritmos básicos y estructuras de datos entre otros.
- Estipulación de criterios para diseñar los fundamentos de proyectos tipo (aplicaciones financieras, sistemas de tiempo real, software de seguridad crítica...)

Y deben quedar definidos cuando se asigna a una persona o a un equipo de personas, la responsabilidad de hacer el diseño.

Para el diseño: herramientas de especificación y para presentación.

Nota: Sobre el diseño se podría escribir mas pero eso, se sale del alcance de este trabajo.

- **Fase operativa.**- Etapa en la que comienzan las actividades programadas del proyecto y se les da el seguimiento preestablecido para que, en caso necesario, se realicen los cambios pertinentes.

Plan de trabajo

Se debe hacer una definición de la forma en que habrá de utilizarse el tiempo del proyecto en las diferentes actividades fundamentales.

Construcción

Se deben definir convenciones que faciliten la lectura de programas escritos por diferentes miembros de la empresa; algunas parámetros que se pueden estandarizar son:

- Métodos de codificación (incluyen las variables, las funciones, presentación y nombramiento de documentos.)
- Conceptos relativos a los datos (incluyendo el ámbito, persistencia y momento de asignación.)
- Directrices para utilizar tipos de datos específicos (incluyendo los números en general, enteros, números con punto flotante, caracteres, cadenas de caracteres, booleanos, tipos enumerados, constantes, vectores y punteros.)

- Conceptos relativos al control (tomando en cuenta la organización lineal del código, uso de condicionales, control de *ciclos*, utilización de expresiones *booleanas*, control de la complejidad y organización de estructuras de control poco usuales, como *goto*, *return* y procedimientos recursivos.)
- Reglas para compactar código dentro de rutinas, módulos, clases y archivos.
- Métodos de depuración y comprobación de unidades.
- Estrategias de integración (como la integración incremental, integración *big-bang* y desarrollo evolutivo.
- Estrategias y métodos para el afinamiento del código.
- Estipulación de lineamientos para escoger el lenguaje de programación adecuado.

Uso de herramientas de construcción (como entornos de programación, soporte para grupos de trabajo como el correo electrónico, y el control de código fuente, bibliotecas de código y generadores de código.)

Control

Se deben definir criterios para dar seguimiento a los proyectos de forma que convenga a los intereses de los mismos y satisfaga las preferencias de Enlaces Tecnológicos, algunos modelos para controlar el avance de los proyectos se enlistan y describen a continuación:

- *Modelo de cascada pura.* En este modelo el proyecto progresa a través de una secuencia ordenada de pasos partiendo del sistema inicial del software hasta la prueba del sistema. Se realiza una revisión final de cada etapa para determinar si se está preparado para pasar a la siguiente etapa; Existen algunas variaciones sobre este modelo de las que las más importantes son entre otras: la cascada con fases solapadas, cascada con subproyectos y la cascada con reducción de riesgos
- *Espiral.* Está orientado a los riesgos y en él se divide un proyecto en subproyectos; cada uno de estos se centra en uno o más riesgos importantes hasta que todos éstos estén controlados.
- *Prototipado evolutivo.* En este modelo se desarrolla el concepto del sistema a medida que avanza el proyecto. Normalmente se comienza desarrollando los aspectos más visibles del sistema. Se puede presentar una parte del sistema al cliente y entonces continuar el desarrollo del prototipo basándose en la retroalimentación que se recibe. Se llega al momento en que el cliente y el equipo de desarrollo en que el prototipo es “lo suficientemente bueno” y en ese momento se completa cualquier trabajo pendiente en el sistema y se entrega el prototipo como producto final.
- *Entrega por etapas.* Con esta estrategia se muestran al cliente etapas refinadas sucesivamente. A diferencia del prototipado

evolutivo, cuando se utiliza la entrega por etapas, se conoce exactamente qué es lo que se va a construir. Lo que hace diferente a este modelo es que el sistema no se entrega al final del proyecto en un solo paso, de golpe!. Se entrega por etapas sucesivas a lo largo del proyecto.

- *Diseño por planeación.* El modelo de diseño por planeación es similar al de entrega por etapas, con la diferencia de que no siempre se conoce al principio si se tendrá el producto para la última entrega. Se pueden tener cinco etapas planeadas, pero sólo se llega a la tercera debido a que se tiene una fecha límite inamovible. Este modelo puede ser una estrategia válida para asegurar que se tiene un producto listo para entregar en una fecha determinada. Si se debe tener absolutamente funcionando al sistema a tiempo para una presentación comercial, o para final de año, o para cualquier otra fecha definitiva, esta estrategia garantiza que se tendrá *algo*. Esta estrategia es particularmente útil para las partes del producto que no se encuentran en la ruta crítica del plan del proyecto.
- *Diseño por herramientas.* Es una radical que históricamente se ha utilizado sólo en entornos sensibles al tiempo. Como las herramientas se han hecho más flexibles y potentes (estructuras de aplicaciones completas, entornos de programación visual, o entornos de programación en bases de datos con prestaciones completas), se ha incrementado el número de proyectos que pueden utilizar el diseño por herramientas. La idea que subyace en este modelo es incluir una prestación dentro del producto sólo si las herramientas de software existentes la soportan directamente. Algunos ejemplos de estas herramientas son: las librerías de código y clases, generadores de código y los lenguajes de desarrollo rápido.

Los aspectos que han de tomarse en cuenta para seleccionar un modelo determinado son entre otros:

- El grado en que están claros los procesos que han de automatizarse o la idea en general del sistema.
- El conocimiento que se tiene del ambiente y los equipos con los que habrá de convivir el nuevo sistema.
- La calidad con la que debe entregarse el sistema.
- Los riesgos conocidos en el proyecto.
- La facilidad que se tiene de planear el proyecto tomando en cuenta la participación del cliente.
- La probabilidad de que existan cambios importantes en medio del proyecto.
- La necesidad de visibilidad en el avance del proyecto.
- La preparación técnica de los clientes.

Pruebas de calidad

Las pruebas de calidad deben abordar los conceptos de modularidad (probar que todos los módulos funcionen), Integración (que el sistema como un todo funcione en conjunto tal como se diseñó), volumen o de estrés (que funcione bien en condiciones extremas de carga de proceso) y de interfaz con el usuario (que sea amigable, es decir, de fácil operación en términos de los que los usuarios finales requieren).

- **Fase de terminación.**- Durante esta etapa se realiza un análisis de los éxitos y fracasos del proyecto (incluyendo su estructura administrativa), para preparar un informe detallado para los equipos de proyectos futuros.

Entrega e instalación

Es un error común que no se plantee la instalación del producto como parte del proyecto mismo, lo que ocasiona retrasos y descontento de los clientes que lo que pretenden es comenzar la operación de su sistema en cuanto éste está terminado. El problema se presenta cuando se debe instalar el sistema en más de un punto de operación, por ejemplo como se hizo en un principio con los cajeros automáticos; así que el plan de instalación debe diseñarse junto con el plan de trabajo, así como la capacitación y la documentación que habrá de entregársele al cliente?

Plan de seguimiento posventas

En un gran número de ocasiones un proyecto entregado con calidad y a tiempo es la llave para realizar nuevos proyectos con un cliente que ya conoce a la compañía y esta relación de largo plazo se ve fomentada por la disposición del proveedor a cuidar del buen desempeño de sus productos y a hacer modificaciones de los mismos si cambian las necesidades del cliente con base en el entendimiento del negocio del cliente.

Asumiendo tras la planeación inicial que el proyecto es beneficioso y viable y por tanto se ha decidido realizarlo, el siguiente paso es identificar las actividades que deben terminarse y las relaciones que existen entre ellas, es de suma importancia realizar esta tarea sin errores dada su inferencia en la programación del proyecto y en la asignación de recursos (el análisis y el diseño detallado son el fundamento para esta tarea). Una herramienta bastante utilizada para identificar actividades que se puedan agrupar en paquetes de trabajo asignables a determinadas personas o equipos de trabajo son las estructuras de desglose de trabajo; Una estructura de desglose de trabajo (que también se puede ver como árbol de objetivos)

separa el proyecto en un conjunto de componentes principales que a su vez se aíslan en componentes cada vez más pequeños, que a su vez pueden ser descompuestos en nuevas estructuras de desglose de trabajo, esto es conveniente cuando el proyecto es complejo y su administrador se auxilia con diferentes responsables para las componentes principales, que a su vez pueden dividirse; Este proceso de desglose está en función de la cantidad y complejidad de actividades necesarias para la terminación del proyecto y del personal involucrado.

También es conveniente utilizar un conjunto de diagramas de red (ya sean con el formato de flecha o con el de nodo) para ilustrar el proyecto haciendo más claras las relaciones entre actividades y facilitar su correcta programación y la elaboración de su presupuesto.

La estimación de tiempo y recursos que se asigna a cada actividad es de la mayor importancia por lo que vale la pena ampliar en el tema:

La estimación es más un arte que una ciencia, es una actividad importante que no debe llevarse a cabo de forma descuidada. Existen técnicas útiles para la estimación de costos y de tiempos; y, dado que la estimación es la base de todas las demás actividades de planeación y que ésta a su vez sirve como guía para una buena administración de proyectos, la estimación de recursos, costos y agendas para el esfuerzo de desarrollo requiere experiencia, acceso a una buena información histórica y aplomo para confiar en medidas cuantitativas cuando todo lo que existe son datos cualitativos. La estimación conlleva *un riesgo inherente*. La **complejidad** del proyecto tiene un gran efecto sobre la incertidumbre, que va aunada con la planeación, sin embargo, es una medida relativa que se ve afectada por la familiaridad con anteriores esfuerzos. El **tamaño** del proyecto es otro factor importante que puede afectar a la precisión y a la eficacia de las estimaciones. A medida que aumenta el tamaño, la interdependencia entre los distintos elementos del proyecto crece rápidamente. El **grado de estructuración** del proyecto también tiene efecto sobre el riesgo de la estimación, a medida que aumenta, la posibilidad de estimar con precisión mejora y el riesgo disminuye. La **disponibilidad de información** también determina el riesgo con que la estimación impactará al proyecto. Se realizan estimaciones para los tiempos, recursos tanto financieros como de humanos y tecnológicos o de información con la idea de que en la medida de que más y mejor información se tenga, se pueden mejorar las estimaciones.

Resumiendo, la estimación del proyecto nunca será una ciencia exacta, pero la combinación de buenos datos históricos y de técnicas sistemáticas puede mejorar la eficacia de la estimación.

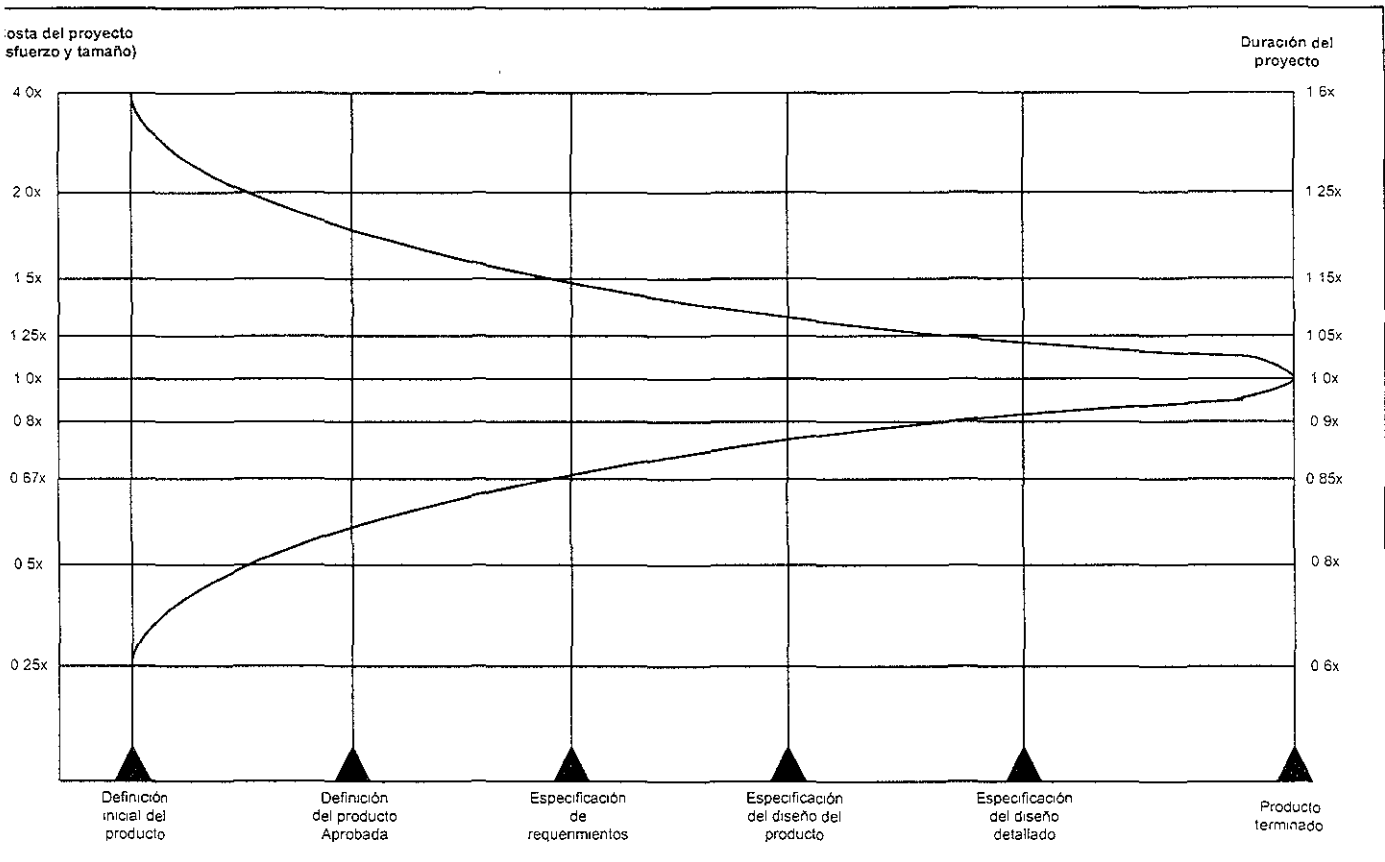
Estimación

"El ideal de una mente instruida es quedar satisfecho con el grado de precisión que admita la naturaleza de un tema, y no buscar la exactitud cuando sólo es posible alcanzar una aproximación de la verdad"

Aristóteles

No se puede estimar con precisión el costo de un proyecto hasta que se comprendan con detalle las características de éste. El desarrollo de un proyecto es un proceso en el que se van tomando decisiones cada vez más detalladas. El concepto de producto se refina en la fase formativa en la que se investigan los requerimientos, que a su vez se depuran con el diseño preliminar, y el diseño preliminar con el diseño detallado y el diseño detallado con la programación de actividades concretas que, finalmente, se depuran con la actividad diaria y las situación que se presentan, la idea es disminuir al máximo esta última.. En cada una de estas etapas se toman decisiones que afectan el costo final y la planeación del proyecto. La incertidumbre sobre la naturaleza del proyecto aporta incertidumbre a la estimación, ya que no se pueden conocer las decisiones que serán tomadas hasta que no se tomen.

A medida que se tiene un avance en el desarrollo del proyecto, son más precisas las estimaciones que falta por hacer, así como un primer intento de hacer una estimación con las generalidades de requerimientos se puede una valer de una gráfica de convergencia para la estimación (COCOMO 2.0 Cost Models Future Life Processes, Boehm 1995) en la que se presentan de forma gráfica los multiplicadores que podrán ser utilizados al realizar las primeras estimaciones, a modo muy general puede resultar una herramienta útil dada la carencia de información con la que se efectúa la primer propuesta del proyecto.



En la gráfica anterior debe entenderse por producto al objetivo del proyecto y la forma en que puede ser utilizada se muestra más claramente en la tabla que a continuación se presenta, que será explicada enseguida:

Fase	Esfuerzo y tamaño		Planeación	
	Optimista	Pesimista	Optimista	Pesimista
Concepto inicial del objetivo del proyecto	0.25	4.0	0.60	1.60
Concepto del proyecto aprobado	0.50	2.0	0.80	1.25
Especificación de requerimientos	0.67	1.5	0.85	1.15
Especificación del diseño del proyecto	0.80	1.25	0.90	1.10
Especificación del diseño detallado	0.9	1.10	0.95	1.05

Para utilizar los factores de la tabla simplemente se multiplica la estimación puntual *más probable* por el factor optimista para obtener la estimación optimista, y por el factor pesimista para obtener la estimación pesimista. La estimación se puede presentar como un rango en lugar de un tiempo puntual. Si la estimación más probable de un proyecto fuera de 50 personas-mes y ya se ha finalizado la especificación de los requerimientos, habría que multiplicar por 0.67 y por 1.5 y estimar un rango de 34 a 75 personas-mes.

Los factores optimista y pesimista para la estimación de la planeación son diferentes a los factores a los del esfuerzo y tamaño; se supone que en la estimación de la planeación, primero se crea una estimación del esfuerzo y luego, a partir de ésta, se calcula la planeación. Si la planeación se ha estimado a ojo, será más conveniente utilizar los rangos mostrados en "esfuerzo y tamaño" que los de la "planeación".

El proceso de estimación se puede dividir en los siguientes puntos :

- Estimar el tamaño del proyecto (número de actividades). Este paso es con mucho el más difícil intelectualmente, y ésta podría ser una de las razones por las que se suele saltárselo y al hacerlo aumenta la probabilidad de fallar en todas las estimaciones consecuentes lo que lleva en general a un fracaso en el plan de trabajo.
- Estimar los recursos (personas - mes,). Si se tiene una estimación exacta del tamaño y de datos previos de organización de proyectos similares, es relativamente fácil realizar la estimación del esfuerzo.
- Estimar el plan (tiempo). Una vez que se han estimado el tamaño y el esfuerzo, estimar el plan resulta casi trivial.

Se puede estimar **el tamaño del proyecto** de varias formas, unas de éstas son:

- Hacer un análisis detallado de todas las actividades necesarias para conseguir un objetivo, estas actividades pueden ser descompuestas en tareas que serán necesarias para realizar dicha tarea. El uso del árbol de objetivos como una herramienta para identificar **todas** las actividades a programar puede ser de gran utilidad.
- Utilizar software de estimación a partir de las descripciones de las actividades a llevar a cabo, este tipo de herramientas se basa en la información reunida a lo largo de mucho tiempo para actividades tipo, como en el caso de la construcción en el que se han estandarizado los tiempos y recursos necesarios para muchas actividades.
- Otra forma de hacerlo, cuando se tiene experiencia previa con proyectos similares, es estimar el nuevo proyecto como un porcentaje del tamaño de un proyecto anterior.

A continuación se presentan algunas recomendaciones que pretenden disminuir el grado de incertidumbre en la estimación del tamaño del proyecto.

Buscar la opinión de expertos, la técnica DELPHI puede ayudar en esta búsqueda.

Realizar estimaciones por consenso, cada miembro del equipo tiene que realizar la estimación de una parte del proyecto de forma individual, y luego en una reunión se comparan las estimaciones, para llegar a una estimación de consenso.

Estimar con el nivel de detalle más bajo posible.

Utilizar técnicas diferentes de estimación y comparar los resultados para estudiar las diferencias y retroalimentar la estimación para hacerla más precisa.

Con la estimación del esfuerzo se hace referencia a los recursos necesarios para llevar a cabo una actividad, tanto de tiempo y de dinero como de personal y equipo. Se debe ahora hacer explícita la dependencia que existe entre actividades para conocer cuáles se deben realizar primero que cuáles.

Ahora, suponiendo que ya se cuenta con la información de las actividades a realizar, su interdependencia y de sus estimaciones tanto de tiempo como de recursos, es posible plantear un calendario de inicio y

terminación de actividades en el que se incluyan los costos asociados con éstas (estimación del plan). Una herramienta que se utiliza con frecuencia es el diagrama de Gantt en el que se describirá y anotará toda la información pertinente (como su duración, responsable, las actividades que la preceden y su costo) de cada actividad, para que en forma gráfica se pueda evaluar el progreso del plan de trabajo, es decir, con ayuda de este tipo de diagramas resulta más fácil la identificación de las actividades que están adelantadas o retrasadas para realizar las actividades convenientes.

Definición.- *el diagrama de Gantt* es un diagrama de barras horizontales en el que la lista de actividades va debajo del eje vertical y las flechas se colocan a lo largo del eje horizontal, en diagramas como estos cada barra representa la duración de una actividad; el principio y final de las barras reproducen las fechas estimadas del comienzo y final de la actividad. las actividades que comienzan más temprano se localizan en la parte superior del diagrama, y las que comienzan después se colocan de modo progresivo, empezando por las que comiencen primero, en el eje vertical.

A continuación es conveniente elaborar un diagrama de red del proyecto para ayudar a los administradores del proyecto a separar y comprender las relaciones entre actividades. Los diagramas de red constan de círculos, cuadros o nodos conectados por flechas, si es éste el formato.

Definición.- En un diagrama de red de actividad en nodo cada actividad se representa mediante un círculo o nodo. Las flechas entre los nodos indican el orden en el que deben ejecutarse las actividades.

Como alternativa, puede trazarse un diagrama de red de actividad en flecha, en el que cada actividad se representa mediante una flecha que conecta dos nodos. En estas redes sólo existe un comienzo y un final para el proyecto. Este tipo de diagramas utilizan comúnmente actividades ficticias para preservar la lógica de precedencia o para asegurar que cada actividad tenga sólo dos nodos; el inicio y el final.

Las herramientas de programación basadas en las redes, como PERT y CPM , pueden emplearse en reemplazo de los diagramas de Gantt o junto con estos. En 1958, la oficina de proyectos especiales de la armada de los Estados Unidos, junto con Booz, Allen y Hamilton, desarrolló la PERT (técnica de revisión y evaluación de programas) para planear y controlar el programa de misiles polaris. En la década de los años cincuenta, Dupont y Remington Rand desarrollaron el CPM (método de la ruta crítica) para ayudar en la construcción y mantenimiento de plantas químicas. En la actualidad, el CPM y la PERT se emplean con frecuencia.

Los diagramas de red PERT y CPM son mejores instrumentos que los de Gantt para ilustrar la secuencia de actividades que deben ejecutarse. Éstos indican cuáles actividades pueden ejecutarse simultáneamente con otras y cuáles no. Otra información que puede deducirse de estas herramientas es la siguiente:

- La fecha estimada de terminación del proyecto
- Actividades que son críticas (que retrasarán el proyecto completo si no se terminan en la fecha indicada)
- Actividades que no son críticas. Estas actividades pueden retrasarse sin que afecten la terminación del proyecto
- Saber si el proyecto está al día, retrasado o adelantado
- Saber si el capital invertido hasta la fecha es igual, mayor o menor que la suma presupuestada
- Saber si existen suficientes recursos disponibles para terminar a tiempo el proyecto
- Conocer la forma más adecuada para reducir la duración del proyecto

Los seis pasos siguientes son comunes en la PERT y en el CPM

- I. Definir el proyecto y todas las actividades o tareas importantes.
- II. Determinar las relaciones entre las actividades. Decidir qué actividades deben preceder a otras y cuáles deben seguir a otras.
- III. Esbozar una red que conecte todas las actividades.
- IV. Asignar tiempo y/o costos estimados a cada actividad.
- V. Calcular el tiempo requerido para completar las actividades en cada trayecto de la red.
- VI. Emplear la red para que sirva de soporte al plan, el programa y el control del proyecto.

Actualmente la única diferencia entre PERT y CPM se relaciona con la duración estimada de cada actividad: CPM emplea un estimado de tiempo de un solo punto de duración de la actividad, mientras que PERT utiliza tres (más probable, probable y menos probable). Los cálculos hoy en día, se hacen auxiliados por software para el dibujo de la red y para los cálculos.

En general, saber la holgura que se tiene en las actividades puede ayudar a los administradores a reprogramar las actividades para moderar los requerimientos de recursos, retrasando lo menos posible el proyecto. También se acostumbra depurar la programación de actividades con base en *la programación del costo mínimo* con la intención de reducir los costos totales del proyecto.

Durante el diseño del programa de actividades se debe realizar un estudio de riesgos con objeto de tener, en la medida de lo posible, planes

establecidos de contingencia. La gestión de riesgos también resulta una actividad en la que se pretende sistematizar lo subjetivo con base en la probabilidad de ocurrencia y en el impacto que tendrían algunas situaciones en caso de presentarse, por supuesto estos valores tienen su fundamento en la experiencia y en la opinión de expertos.

Otro aspecto que vale la pena resaltar en la administración de un proyecto es la búsqueda, organización y control del equipo de trabajo, para algunos proyectos en específico ya se han hecho estudios que proporcionan información validada por la experiencia de muchos usuarios, lamentablemente en otro tipo de proyectos como en el del software, la tecnología cambia tan rápido que resulta complicado pensar en estándares que faciliten la estimación del tiempo que tardará un integrante de un equipo de trabajo en hacer alguna tarea.

Resulta de la mayor utilidad, dado que a final de cuentas el plan de actividades se basa en estimaciones sujetas a posibles errores, tomar en cuenta los posibles planes de prevención y contingencia que se emplearán para evitar retrasos o que se emprenderán en caso de que se presente alguna desviación del plan, respectivamente. Esta actividad es conocida como la **gestión de riesgos**, que involucra la estimación de riesgos y el control de los mismos, y en la actualidad ha cobrado un gran auge, la gestión de riesgos, también, se puede dividir en los siguientes niveles:

- I. Control de crisis.- Apagar el fuego, controlar los riesgos sólo cuando se han convertido en problemas
- II. Arreglar cada error.- Detectar y reaccionar rápidamente ante cada riesgo pero sólo hasta que se producido
- III. Mitigación de riesgos.- Planificar con tiempo qué se necesitaría para cubrir riesgos en el caso de que ocurran, pero no se intenta eliminarlos antes de que ocurran
- IV. Prevención.- Crear y llevar a cabo un plan como parte del proyecto para identificar riesgos y evitar que se conviertan en problemas
- V. Eliminación de las causas principales.- Identificar y eliminar los factores que puedan hacer posible la presencia de algún tipo de riesgo

La estimación de riesgos se compone de identificación de riesgos, análisis de riesgos y asignación de prioridades a los riesgos.

- La identificación de riesgos genera una lista de riesgos capaces de romper con el plan del proyecto
- El análisis de riesgos.- Mide la probabilidad y el impacto de cada riesgo, y los niveles de riesgo de las acciones alternativas

- La asignación de prioridades a los riesgos.- genera una lista de riesgos ordenados por su impacto. Esta lista sirve como base para el control de riesgos

El control de riesgos se compone de la planeación de la gestión de riesgos, resolución de riesgos y monitorización de riesgos.

- La planeación de la gestión de riesgos genera un plan para tratar cada riesgo significativo. También asegura que los planes para la gestión de cada uno de los riesgos individuales son consistentes entre sí y con el plan del proyecto
- La resolución de riesgos es la ejecución del plan para resolver cada uno de los riesgos significativos
- La monitorización de riesgos es la actividad del progreso de la monitorización dirigido a la resolución de cada elemento del riesgo. La monitorización de riesgos también puede incluir la continuación de la actividad de la identificación de nuevos riesgos y volver a considerarlos en el proceso de la gestión de riesgos

La identificación de riesgos es una tarea que requiere de gran experiencia en la elaboración de proyectos debido a que es necesario hacer una lista de los posibles riesgos lo más exhaustivamente posible, los riesgos pueden ser originados por cualquiera de los participantes en el proyecto o por su entorno, se debe pensar en TODOS! Los posibles riesgos que involucren, por mencionar a algunos, los siguientes: El plan de trabajo, la organización del proyecto, el entorno en el que se llevará a cabo el proyecto, los usuarios finales, los clientes, el personal contratado, los requisitos del proyecto y fuerzas mayores entre otros.

Una vez que se han identificado los riesgos en el plan del proyecto, el paso siguiente es analizar cada riesgo para determinar su impacto. Existen diferentes herramientas que intentan facilitar esta análisis un ejemplo es el denominado método de exposición al riesgo (ER) y otro es el análisis de impactos cruzados. La idea es determinar qué pasaría si se presentara determinado riesgo con base en su probabilidad de ocurrencia y en su impacto (magnitud de la pérdida, ya sea en tiempo o en costo) para asignarle una prioridad a cada uno.

La exposición al riesgo se calcula multiplicando la probabilidad de ocurrencia del riesgo por la magnitud de la pérdida.

Estimación de la magnitud de la pérdida muchas veces se basa en información que depende de las características especiales del proyecto, por ejemplo si el arranque del proyecto depende de que sea aprobado por el Consejo de Administración (C.A.) de una compañía que sesiona una vez al mes, se sabe que la pérdida por no entregar la documentación adecuada

del proyecto para su evaluación ante el C.A. provocaría un retraso de un mes, por supuesto, otras muchas veces también es necesario estimar la magnitud de la pérdida con base en la experiencia de los participantes o en la opinión de expertos en el tema.

La estimación de la probabilidad de ocurrencia es más subjetiva todavía, así que a continuación se presentan algunas ideas para mejorar la exactitud de la estimación.

- Disponer de la persona que esté más familiarizada con el proyecto o con proyectos del mismo tipo para que estime la probabilidad de cada riesgo, y luego realizar una revisión de la estimación.
- Utilizar técnicas DELPHI o de consenso en grupo. Con la técnica DELPHI, cada persona (de preferencia expertos en el tema) estima individualmente cada riesgo, y luego se discute o se escribe el origen de cada estimación, especialmente en las que haya mayores diferencias. Continuar con el proceso hasta que exista consenso.
- Realizar analogías con apuestas: "¿aceptarías esta apuesta? si los recursos están disponibles en su momento ganas \$125. Si no lo están yo gano \$100". Ajustar la apuesta hasta que los dos apostadores estén igual de seguros. La probabilidad de riesgo sería la apuesta menor dividida entre la cantidad total en juego.

La información del análisis de riesgos puede ser utilizada para replantear el programa de trabajo, teniendo ahora un margen de error mas cuidadosamente estimado.

Una vez que se ha creado la lista de los riesgos en el plan, el paso siguiente es priorizar los riesgos de forma que sepa dónde centrar los mayores esfuerzos para la gestión de riesgos. Los proyectos gastan generalmente el 80% de su presupuesto en arreglar el 20% de sus problemas, por lo que es útil poder centrarse en este 20% más importante.

La resolución de un riesgo concreto depende mucho de la naturaleza de éste. Así que nuevamente sólo se presentan algunos consejos útiles que en general se puede aplicar en los proyectos de una gran variedad de tipos.

- Traslade, de ser posible, el riesgo de una parte del proyecto a otra. A veces lo que es un riesgo en una parte del proyecto no lo es en otra.
- Consiga toda la información posible del riesgo.
- De ser posible elimine la raíz del riesgo.
- Comunique a todos los participantes (pertinentes) que el riesgo existe y sus posibles consecuencias.
- Crear una colección de riesgos y sus consecuencias para que queden como memoria para futuros proyectos.

Como por lo general no pasa que los riesgos aparezcan después de que se haya desarrollado un plan de contingencia, es necesario llevar un estudio de los riesgos que se presentan, cosa que por lo general es dinámica, es decir, que van apareciendo y desapareciendo mientras dura el proyecto, en el que se resalten los diez riesgos de mayor relevancia en un determinado tiempo, una de las herramientas más utilizadas para este objetivo es la llamada lista de los diez riesgos creada por el administrador del proyecto, en la que se muestra la posición del riesgo en la lista, su posición anterior, el número de veces que ha aparecido en la lista y un resumen de las acciones que se han llevado a cabo para controlarlo. No es indispensable que sean exactamente diez los riesgos en la lista.

Construcción

Después de generar planes de trabajo con objetivos y tiempos definidos, de hacer una gestión de riesgos adecuada y todas las demás actividades de planeación para el proyecto, se llega el momento en que se construirá el sistema, para esto existen un sin número de métodos, convenciones, herramientas y metodologías, en particular se describirá a continuación el sistema CVS que será utilizado para controlar las versiones de un desarrollo cuando se trabaja en equipo o a distancia.

CVS es un sistema de control de versiones. Usándolo se puede almacenar la historia de los cambios en los archivos fuente.

Por ejemplo, a veces los *bugs* (Errores en sistema) cuando éste es modificado y por lo general se detectan mucho tiempo después de que se hizo el cambio que lo causó. Con el CVS se puede reconstruir la versión anterior a la aparición del inicio del *bug* y saber exactamente qué cambios le fueron hechos lo que puede resultar de gran ayuda.

Por supuesto se puede salvar cada versión de cada archivo que se haya creado, pero sería un enorme desperdicio de espacio en el disco. Los archivos en CVS almacenarán todas las versiones de cada archivo del sistema que se desarrolla de forma eficiente (astuta).

El CVS también resulta de gran ayuda cuando trabajan muchas personas en un mismo proyecto, normalmente es muy fácil que en esta situación se sobrescriba el trabajo y los cambios de unos y de otros, lo que obliga a ser extremadamente cuidadoso y por consiguiente hacer el trabajo en más tiempo. Algunos editores como el GNU Emacs están orientados a tratar de asegurarse de que el mismo archivo nunca sea modificado por dos personas al mismo tiempo. Desafortunadamente si alguien está utilizando otro editor ese mecanismo de seguridad no funciona. El CVS resuelve este problema aislando a cada desarrollador de los otros. Cada desarrollador

trabaja en su propio directorio y el CVS acopla el trabajo cuando éste está terminado.

El CVS surgió como un conjunto de programas conectados desarrollados por Dick Grune, publicados en el *newsgroup* comp.sources.unix, en el volumen 6 realizado en diciembre de 1986.

Se puede tener el CVS de varias formas, incluyendo la descarga gratuita de la Internet.

El sistema CVS se encargará, básicamente, de:

- Asignar al usuario una copia de la última versión del sistema.
- Presentar los cambios efectuados y actualizar la versión.
- Verificar que no existan conflictos y si los hay señalarlos.
- Mantener a la vista los cambios entre las diferentes versiones.

El sistema CVS tiene como partes principales un repositorio, que almacenará una copia completa de los directorios y archivos que estarán bajo control de versiones. Normalmente, nunca se accesa algún archivo directamente del repositorio. De hecho, se utilizan los comandos CVS para obtener una copia en el *directorio de trabajo*, y se trabaja sobre esa copia. Cuando se ha terminado el conjunto de cambios se devuelven al repositorio. Éste ahora guarda los cambios que se han hecho, así como la persona que los hizo, la descripción de ellos y la fecha en que fueron efectuados junto con la información que el desarrollador considere pertinente. Nota: el repositorio no es un subdirectorio del directorio de trabajo, o viceversa, y estarán almacenados en diferentes localidades de memoria.

El repositorio está dividido en dos partes fundamentales, una que contiene los archivos administrativos de CVS y otra en la que se guardan los módulos en los que trabaja el usuario.

Un usuario de CVS debe saber:

- Cómo se guardan los datos en el repositorio.
 - Dónde son almacenados los datos dentro del repositorio.
 - Los permisos de cada archivo.
 - Cómo es el directorio CVS dentro del repositorio.
 - La seguridad del CVS en el repositorio.
 - Cómo son almacenados los datos en el *directorio raíz* del sistema CVS.
- Cómo se guardan los datos en el directorio de trabajo.
- Cómo son los archivos administrativos.
 - Cómo editar los archivos administrativos.
- Cómo se trabaja con repositorios múltiples.

- Cómo crear un repositorio.
- Cómo respaldar un repositorio.
- Cómo mover un repositorio.
- Cuando se trabaja con un repositorio remoto:
 - Requerimientos del servidor.
 - Cómo conectarse con un *rsh*.
 - Cómo establecer un conexión directa con un *password* de autenticación.
- Cómo restringir el acceso de la forma *sólo lectura*.
- Cómo deben ser los directorios temporales en el servidor.

Ahora bien, lo primero que debe hacerse para desarrollar un proyecto con ayuda del sistema CVS es pensar en la administración de los archivos. No es imposible renombrar y mover archivos pero esto en general incrementa la posibilidad de confusión y la aplicación podrá tener problemas para bautizar archivos. Así que resulta conveniente con forma de árbol con los números de archivos.

También desde un comienzo se deben especificar las reglas que seguirá el sistema para hacer las revisiones entre versiones que presenten conflictos entre cambios.

Los responsables de desarrollar un sistema trabajando con la ayuda del CVS deberán conocer cómo han de interactuar con el sistema.

Por cuestión de espacio no se describe más a detalle el funcionamiento del sistema CVS pero existen manuales que se pueden consultar el línea en Internet.

Algunas estrategias para fijar el criterio con que se llevará el control son:

- Modelo de cascada pura.- En este modelo el proyecto progresa a través de una secuencia ordenada de pasos partiendo del sistema inicial del software hasta la prueba del sistema. Se realiza una revisión final de cada etapa para determinar si se está preparado para pasar a la siguiente etapa; Existen algunas variaciones sobre este modelo de las que las más importantes son entre otras: la cascada con fases solapadas, cascada con subproyectos y la cascada con reducción de riesgos.
- Espiral.- Está orientado a los riesgos y en él se divide un proyecto en subproyectos; cada uno de estos se centra en uno o más riesgos importantes hasta que todos éstos estén controlados.
- Prototipado evolutivo.- En este modelo se desarrolla el concepto del sistema a medida que avanza el proyecto. Normalmente se comienza desarrollando los aspectos mas visibles. Se puede presentar una

parte del sistema al cliente y entonces continuar el desarrollo del prototipo basándose en la retroalimentación que se recibe.

- Entrega por etapas.- En este modelo se muestra el software al cliente en etapas refinadas sucesivamente. A diferencia del Prototipado evolutivo, cuando se utiliza la entrega por etapas, se conoce exactamente qué es lo que se va a construir cuando se procede a hacerlo. Lo que diferencia a este modelo es que el programa no se entrega al final del proyecto en una sola entrega final. Se entrega por etapas sucesivas a lo largo del proyecto. Este modelo se conoce también como *implementación incremental*.
- Diseño por planificación.- Este modelo también es similar a la entrega por etapas, en la que se planea desarrollar el producto en etapas sucesivas. La diferencia radica en que no siempre se conoce al principio si se tendrá el producto para la última entrega. Se pueden tener cinco etapas planeadas, pero sólo se llega a la tercera porque se tiene una fecha límite inamovible. Este modelo puede ser una estrategia válida para garantizar que se tiene un producto listo en una fecha determinada.
- Diseño por herramientas. Es una aproximación radical que históricamente se ha utilizado sólo excepcionalmente en entornos sensibles al tiempo. Como las herramientas se han hecho más flexibles y potentes (estructuras de aplicaciones completas, entornos de programación visuales y entornos de programación en bases de datos con prestaciones completas entre otros), se ha incrementado el número de proyectos que pueden utilizar el diseño por herramientas. La idea fundamental de este modelo es incluir una prestación dentro del producto sólo si las herramientas de software existentes la soportan directamente. En caso contrario, se deja. Por *herramientas* me refiero a librerías de código y clases, generadores de código, lenguajes de desarrollo rápido y otras herramientas de programación que reducen el tiempo de implementación de forma espectacular.

La decisión de adoptar una u otra estrategia estará en función de los aspectos particulares de cada proyecto.

Ya que se ha construido y probado un sistema se debe hacer un esfuerzo importante para generar los manuales de usuario y técnicos que habrán de entregarse al cliente. Además se debe planear la forma en que se pondrá en marcha el nuevo sistema, porque en numerosas ocasiones un sistema deberá de sustituir a otro que está en operación sin interrumpirla y éste es un problema que merece una atención especial.

Por lo general el compromiso de una proveedora de sistemas informáticos termina al entregar el sistema que se la ha solicitado funcionando tal y

como se estipuló; y es responsabilidad del cliente su uso, aunque el proveedor debe capacitar a los usuarios y darles la información técnica necesaria para operar el sistema.

Cuando se instala un sistema que será utilizado para automatizar un proceso que no ha sido automático nunca o que está parado o que no existía, la puesta a punto es relativamente fácil. Para que opere el sistema debe contener la información con la que operará y la actividad de capturar esa información puede requerir de mucho tiempo y personal, como en el caso de que se tenga que dar de alta en el sistema a todos los usuarios de una biblioteca o las placas de los autos con permiso de entrar al estacionamiento de una empresa grande.

Ahora, si el sistema que ha de sustituirse está en operación y no se debe de suspender ésta en el cambio, se debe tener el sistema a punto antes de iniciar su uso, se deben sincronizar y muchas veces correr al mismo tiempo para hacer la migración, si el sistema es muy grande el equipo que se usará tendrá que ser duplicado, lo que representa un gasto importante.

Si además el sistema será instalado en diferentes puntos de operación y éstos son muchos tal vez sea necesario hacer un modelo de transporte para hacer la programación de la instalación de forma óptima.

Estos y otros muchos factores que cambian de uno a otro proyecto deben ser tomados en cuenta al hacer el plan del proyecto ya que su omisión repercute en la satisfacción del cliente y en futuras oportunidades de negocio.

El resumen anterior fue obtenido del trabajo de diferentes autores que han realizado esfuerzos por fomentar la cultura de la administración de proyectos informáticos, por supuesto que algunos tratan con mayor profundidad algunos temas que otros e incluso hay temas que no tratan y otros sí. Por lo anterior a continuación se presenta una tabla en la que se hace referencia a los puntos que los diferentes autores tomados en cuenta para este trabajo haciendo una comparación entre ellos.

Autor	Temas de la administración de proyectos informáticos que tratan						
	Análisis	Diseño	Gestión de Riesgos	Planeación	Calidad	Organización	Control
McConnell	X	X	X	X	X	X	X
De Marco		X	X	X	X		X
Glib	X	X	X	X	X		X
Senn	X	X			X	X	
Sierra	X	X					
Metzger	X	X		X	X		X

Otros temas importantes son:

Autor	Temas de la administración de proyectos informáticos que tratan						
	Modelos de ciclo de vida	Estimación	Motivación	Equipos de trabajo	Productividad	Entrega e instalación	errores
McConnell	X	X	X	X	X		X
De Marco		X			X		X
Glib				X			
Senn	X	X			X	X	
Sierra	X	X					
Metzger			X	X	X		X

Apéndice II Motivación

Las personas ofrecen la mayor posibilidad de acortar la duración de un plan de para el desarrollo de software a lo largo de una serie de proyectos. La mayoría de las personas que trabajan en la creación de sistemas informáticos han observado personalmente la gran diferencia en el resultado del trabajo de desarrolladores medios, mediocres y genios. La motivación es indudablemente la mayor influencia individual sobre cómo trabajan las personas. La mayoría de los estudios de productividad han encontrado que la motivación tiene mayor influencia en la productividad que ningún otro factor, en especial cuando el trabajo ha realizarse es básicamente intelectual (Boehm, 1981).

Considerando el papel crítico que juega la motivación en la velocidad de desarrollo, sería de esperar tener un programa de motivación completo ocupando una posición de gran importancia en todo proyecto de desarrollo. Pero ése no es el caso. **La motivación es un factor intangible:** difícil de cuantificar, y suele considerarse después de otros factores que podrían ser menos importantes, pero que son mas fáciles de medir. Muchos métodos de gestión comunes hacen economías ridículas y luego grandes desperdicios, cambiando enormes pérdidas en la moral para mejorar levemente la metodología o lograra ahorros dudosos en el presupuesto.

Personas diferentes se motivan con factores diferentes, y por lo general los desarrolladores no se motivan siempre con los mismos factores que sus directivos ni con los mismos factores que las personas en general. Las comparaciones entre los aspectos que motivan a los desarrolladores con los de los directivos son particularmente interesantes, y nos ayudan a explicar algunos de los motivos de la falta de comunicación que existen entre ellos. *Si un directivo intenta motivar a sus desarrolladores de la misma forma que le gustaría que lo motivaran a él, realmente fallará.* Los desarrolladores se preocupan poco de los valores que caracterizan a los directivos como el incremento en la responsabilidad o la posibilidad de coordinar grande áreas de las empresas sean de sistemas informáticos o no. Si se desea motivar desarrolladores debe hacerse hincapié en los estímulos técnicos, autonomía, posibilidad de aprender, utilización de nuevas técnicas, tomar en consideración su opinión y vida personal en la planeación del proyecto.

En algunos estudios se han identificado cinco factores de motivación que a los desarrolladores de software especialmente les interesan:

- **Realización.** A los desarrolladores de software les gusta trabajar. La mejor forma de motivar a los desarrolladores es proporcionarles un entorno que les facilite centrarse en lo que más les gusta hacer, que es desarrollar software.
 - *Propiedad.* La propiedad (“comprar”) es una clave para la motivación por realización. Las personas trabajan más duro para alcanzar sus propios objetivos que para alcanzar los de otras personas. Se ha visto que si se tiende a dejar que los desarrolladores creen sus propios planes, se facilita la planeación de proyectos grandes (de mucho tiempo) porque los planes de los desarrolladores siempre son ambiciosos. El directivo deberá preocuparse de no tener estimaciones demasiado optimistas o por los desarrolladores que trabajan voluntariamente demasiadas horas extras, pero esos no son problemas de motivación.
 - *Definición de objetivos.* Establecer los objetivos de la velocidad de desarrollo explícitamente es un paso obvio y simple para alcanzar el desarrollo acelerado de software, y resulta fácil de supervisar. Podría preguntarse que si se establece un objetivo de tiempo de desarrollo, ¿trabajarán realmente los desarrolladores para conseguirlo? Si conocen cómo cuadra ese objetivo con otros y si el conjunto de objetivos es razonable, la respuesta es que sí. Los desarrolladores no pueden responder de los objetivos que cambian diariamente o que son imposibles de cumplir en su conjunto. Los objetivos deben ser claros, aunque no necesariamente simples para funcionar. También definir demasiados objetivos al mismo tiempo es un problema común. La respuesta más común cuando se pregunta a los equipos de desarrollo ¿qué hace que el responsable del proyecto impida que el equipo funcione con mayor efectividad? Es que el responsable diluye los esfuerzos del equipo con demasiadas prioridades en un gran número de objetivos. De lo que se aprende que se deben seleccionar los objetivos y dejar bien claro el orden de importancia de cada uno con respecto del plan y su progreso.
- **Posibilidad de superación.** Uno de los aspectos más excitantes de un desarrollador de software es saber que se trabaja en un campo que está cambiando constantemente. Cada día hay que aprender algo para estar actualizado, y la mitad de lo que hoy se necesita saber para realizar su trabajo estará anticuado dentro de tres años. Y dada la naturaleza de la industria elegida por los desarrolladores no es de sorprender que estén motivados por las posibilidades de superación. Una organización puede mostrar interés en el progreso profesional de sus desarrolladores de distintas formas, entre ellas las más comunes son:

- Ofreciendo dinero para cursos de desarrollo profesional.
- Dando tiempo extra para asistir a clases o estudiar.
- Ofreciendo dinero para la compra de libros profesionales.
- Asignando a los desarrolladores a proyectos que aumentarán su experiencia.
- Asignando un tutor a cada desarrollador nuevo (mostrando tanto al tutor como al nuevo desarrollador que la organización se interesa por la formación de su personal).
- Evitando presiones excesivas en la planeación (indicando a los desarrolladores que la prioridad principal y real es entregar el nuevo producto, independientemente del costo de personal).

En resumen apoyar el desarrollo profesional es vital para la estabilidad de cualquier empresa, y sobretodo en el campo del desarrollo de software.

- **El trabajo en sí.** Generalmente, la motivación interna (en sentido laboral) de las personas parte de tres fuentes: su trabajo debe tener sentido; debe tener responsabilidad por el resultado del trabajo; y debe conocer los resultados reales de las actividades de su trabajo. Se han identificado cinco dimensiones del trabajo en sí que contribuyen a estas fuentes de motivación. Las tres primeras características de este trabajo contribuyen a entender el significado que las personas encuentran en su trabajo:
 - *La diversidad técnica.* Es el grado en el que el trabajo requiere que ejercite una serie de técnicas, de forma que pueda evitar el aburrimiento y la fatiga por rutina. Las personas encuentran significado en aquellos trabajos que ofrecen variedad, incluso en aquellos trabajos que no son muy significativos ni importantes en un sentido absoluto.
 - *La identidad de la tarea.* Es el grado en que su trabajo requiere que cubra una parte completa, identificable, de su trabajo. Las personas ponen más interés en su trabajo cuando éste está acotado claramente y sienten que su colaboración es importante.
 - *La importancia de las tareas.* Es el grado en el que trabajo afecta a otras personas y contribuye al bienestar social. “*las personas que aprietan las tuercas de los aviones sentirán que su trabajo es más importante y significativo que las personas que aprietan las tuercas de los espejos de decoración*”. Del mismo modo, los desarrolladores a los que se les permite reunirse con los clientes y comprender el marco global dentro del cual realizan su trabajo, es posible que se sientan más motivados por su trabajo que si se mantienen en el anonimato. Por supuesto que siempre hay

excepciones, habrá personas que el hecho de tener que entenderse con los clientes le moleste.

La cuarta característica del trabajo contribuye al sentimiento de responsabilidad de la persona sobre el resultado del trabajo:

- *Autonomía.* Es el grado en el que tiene control sobre los medios y métodos que utiliza para realizar su trabajo, el sentimiento de ser su propio jefe, y el margen de maniobra del que dispone. Cuanto mayor sea la autonomía que tienen las personas, mayor es el sentimiento de responsabilidad personal que tienden a sentir por el resultado de su trabajo.

La quinta y última característica del trabajo contribuye al conocimiento de las personas sobre los resultados reales de las actividades de su trabajo:

- *Retroalimentación del trabajo.* Es el grado en el que la realización del trabajo en sí proporciona información clara y directa sobre cómo es de efectivo. (Esto es diferente de recibir la retroalimentación por parte de un supervisor o un colaborador.) El desarrollo del software proporciona una gran retroalimentación en el trabajo porque el trabajo en sí (el programa o sistema) la proporciona: se puede ver inmediatamente si el programa funciona cuando se ejecuta.

Una clave para la motivación es controlar estas cinco dimensiones para crear un trabajo significativo, y posteriormente asignar este trabajo a personas que tienen un gran deseo de realizarlo. La importancia de la calidad en el plan de trabajo es una de las razones que normalmente motiva más a los desarrolladores. Crear algo en una tecnología de punta es un impulso para una persona que está orientada técnicamente. Proporciona un nivel de motivación que es difícil crear fuera de un equipo de proyecto que está mejorando el estado del arte de una u otra forma.

- *Oportunidad para centrarse en el trabajo en sí.* Otro aspecto motivacional del trabajo es el grado en que el entorno permite al desarrollador centrarse en su trabajo exclusivamente, comparado con la cantidad de tiempo que necesita para centrarse en otros temas relacionados.

En la mayoría de las organizaciones los desarrolladores pasan una parte significativa del día realizando tareas administrativas triviales que les hacen perder el día en una serie de distracciones que no contribuyen al proyecto en el que trabajan. Por ejemplo para

conseguir materiales de oficina es necesario hacer un gran número de trámites, o para sacar fotocopias o hacer impresiones se deben programar con tiempo de anticipación y esperar a que sean generadas durante la noche, también es común que en caso de presentarse fallas en el equipo de cómputo se deba esperar mucho tiempo a que sea reparado o lo debe reparar el usuario. Para un desarrollador de software que no desea más que realizar su trabajo, puede ser increíblemente frustrante (y desmotivador) tener que perder el tiempo rellenando un formulario para conseguir un bloc de notas.

Además de no eliminar los trastornos administrativos, algunas empresas tradicionales distraen inadvertidamente la atención del trabajo en sí poniendo más énfasis en aspectos de su entorno, fuera del trabajo. Imponer un tipo de vestimenta sugiere que el trabajo en sí, aunque es importante, no es lo más importante. Imponer un horario de trabajo estricto sugiere lo mismo. En algunas organizaciones, tales políticas son beneficiosas para la imagen de la organización, pero toda organización debiera asegurarse de comprender el mensaje que todas las políticas ajenas al trabajo en sí transmiten a los desarrolladores. Valdría la pena preguntarse si la imagen es suficientemente importante como para perder motivación y productividad.

- **Vida personal.** Realización, posibilidad de superación y el trabajo en sí están entre los cinco primeros factores de motivación, tanto para los desarrolladores como para los directivos (aunque les asignen diferentes prioridades). Esos factores presentan una oportunidad significativa para los desarrolladores y directivos de comprender la función que realizan los demás. Pero la vida personal está en cuarto lugar para los desarrolladores, y en quinto para los directivos. Así, el impacto motivacional de la vida personal de un desarrollador es probablemente el factor más duro de comprender por un directivo. El segundo es probablemente la responsabilidad, que está en primer lugar para los directivos y en el décimo para los desarrolladores.

Un resultado de esta disparidad es que los directivos premien en ocasiones a sus mejores desarrolladores asignándoles sus proyectos más complejos y de mayor presión. Para los directivos, la responsabilidad extra sería un placer y la disminución de la vida personal no importaría mucho. Para un desarrollador, la responsabilidad extra sería más un engaño que un placer, y la disminución de la vida personal una fuerte pérdida. El desarrollador interpreta el *premio* del directivo como un castigo.

Una compañía no puede hacer mucho por utilizar las vidas personales como factores de motivación, excepto planear los proyectos de forma realista, de manera que los desarrolladores

tengan tiempo para su vida personal, respetando periodos de vacaciones y fiestas, y siendo susceptible a las peticiones de tiempo libre durante el trabajo.

- **Oportunidad de supervisión técnica.** Los directivos están menos motivados por la oportunidad de supervisión técnica que los desarrolladores. La forma más fácil de comprender esto es reconocer la conexión entre la oportunidad de la supervisión técnica y la realización. Para un desarrollador, una oportunidad de superación técnica representa una realización. Una oportunidad de un trabajo de supervisión técnica implica que el desarrollador ha alcanzado un nivel de experiencia técnica suficiente para dirigir a otros. Para un directivo, esta misma oportunidad quizá representaría un paso atrás; el directivo ya está supervisando a otros, y en general se encuentra bastante contento de no tener que supervisar los detalles técnicos.

Las oportunidades de supervisión técnica no se limitan a asignar a una persona como responsable técnico de un proyecto. Se puede usar este factor de motivación en términos más generales:

- En un proyecto, se puede asignar a cada persona como responsable técnico de un área del producto en particular (diseño de interfaz con el usuario, formato de informes, operaciones de base de datos, gráficos, impresiones, análisis, redes, comunicación con otras aplicaciones e instalación entre otros muchos).
- Asignar a cada una de las personas la responsabilidad técnica de un área del proceso en particular (revisiones técnicas, reutilización, integración y evaluación de desempeño del sistema por mencionar algunos).
- Hacer que todos los desarrolladores, menos los recién llegados, sean monitores. Puede hacer que monitores de segundo nivel trabajen con monitores de primer nivel. Los monitores de segundo nivel pueden ayudar con las propias actividades de monitoreo, o podrían aconsejar de forma más técnica a los monitores de primer nivel.

Algunas empresas intentan motivar a sus equipos de trabajo con premios e incentivos como:

- Pagos extraordinarios.
- Cenas de reconocimiento con ejecutivos de la empresa.
- Gratificación de vacaciones.
- Obsequios de agradecimiento.
- Y ceremonias de premiación entre otros.

Los premios son una importante fuente de motivación a largo plazo. Pero los premios de dinero deben darse con cuidado. Los desarrolladores por lo general son buenos con las matemáticas, y pueden calcular cuando un premio no es proporcional al sacrificio que han realizado. Se ha visto que en malos sistemas de recompensa la organización aumenta un 6% a trabajadores que han hecho un esfuerzo importante y un 5% a trabajadores mediocres, lo que provoca que trabajadores excelentes se frustren y dejen la compañía. También se ha presentado en numerosas ocasiones que un mal sistema de estímulos permita que se llegue al extremo en el que los premios se vuelven parte del salario o provocan descontento y problemas laborales severos. Algunas acciones que han dado resultado son:

- Una felicitación sincera por un logro específico.
- Camisetas, relojes, jarras, carteles y demás distintivos del equipo.
- Premios en broma o en serio en forma de placas, certificados, trofeos y cosas por el estilo.
- Acontecimientos especiales para conmemorar logros significativos; dependiendo de las preferencias del equipo.
- Cursos especiales.
- Y asenso de categoría entre otros.

Como mencionó Alfie Kohn *“¿los premios motivan a las personas? En absoluto. Motivan a las personas para que consigan premios.*

No existe un procedimiento rígido que alguna empresa pueda seguir para motivar a sus empleados dado lo variable, de persona a persona, que pueden ser los factores que la mejoren, se deben hacer esfuerzos importantes para mejorarla tomando en cuenta la participación de los implicados que son por lo general los que mejor conocen lo que puede mejorarles el interés por un proyecto específico.

También existen factores que van en detrimento de la motivación y que son comunes en la mayoría de las personas, a saber:

- *Factores de higiene.*- Son las condiciones básicas que un trabajador necesita para realizar sus actividades efectivamente. En el mejor caso, su ausencia crea descontento. Una iluminación adecuada es un factor de higiene, porque si la iluminación no es la adecuada, se deteriora la capacidad del trabajador de llevar a fin sus tareas con eficacia, y esto perjudica la motivación. Pero iluminar más de lo necesario no mejora la motivación. A continuación se lista una serie de factores de higiene que son importantes en especial para los desarrolladores de software:

- Iluminación, calefacción y aire acondicionado apropiados.
 - Espacio adecuado para la mesa de despacho y la estantería.
 - Tranquilidad suficiente para poder concentrarse (incluyendo la posibilidad de disminuir el teléfono).
 - Intimidad suficiente para evitar interrupciones no deseadas.
 - Acceso a equipo de oficina (fotocopiadores o fax entre otros).
 - Acceso rápido a los suministros de oficina.
 - Acceso no restringido a la computadora.
 - Equipo informático actualizado.
 - Reparación inmediata o en el menor tiempo posible de los equipos dañados.
 - Soporte actualizado de comunicaciones (incluyendo e-mail, teléfonos individuales, buzón de voz y acceso a salas de conferencias bien equipadas).
 - Herramientas de software pertinentes (tratamiento de textos, herramientas de diseño, editor de programas, compilador, bibliotecas de código y ayudas de depuración por mencionar algunas).
 - Hardware pertinente (por ejemplo, una impresora en color si se está trabajando con aplicaciones gráficas y los clientes esperan que se escriba en color).
 - Manuales de referencia y publicaciones pertinentes.
 - Libros de consulta y ayudas de referencia en línea.
 - Formación mínima en software, metodología y herramientas (la formación adicional puede ser un motivador de la superación).
 - Copias legales de todo el software utilizado.
 - Libertad de establecer el horario de trabajo, tanto en general (escoger trabajar de 8 a 5 ó de 11 a 8, ó en otras horas) como específicamente.
- *Manipulación de la directiva.*- Algunos directivos intentan manipular a los desarrolladores estableciendo unas fechas límites falsas, y en la mayoría de los casos siempre se sabe de la farsa y por supuesto resulta en una baja en la motivación.
 - *Presión excesiva de la planeación.*- Una de las formas más rápidas de bajar la motivación a cero es presentar a los desarrolladores una fecha límite imposible. Pocas personas trabajarán duro para alcanzar una fecha límite que saben que es imposible.
 - *Falta de apreciación de los esfuerzos de desarrollo.*- Las personas, por lo general, no ven la cantidad de trabajo que los desarrolladores están realizando, de forma que piensan que no están haciendo mucho y, por tanto, deben obligarlos. En realidad, los desarrolladores están extremadamente automotivados, trabajan duro y muchas horas, y encontrarse acusados de *holgazanear* es muy desalentador. Si se desea que los desarrolladores hagan más

de lo que realmente ponen de manifiesto en la oficina, nunca jamás hay que insinuar que no están trabajando duro, cuando sí lo están haciendo.

- *Participación de directivos sin preparación técnica.* Los desarrolladores pueden ser motivados por directivos técnicamente incapaces, siempre que esos directivos reconozcan que no tienen preparación técnica y limiten el control del proyecto a decisiones no técnicas. Si intentan meterse en decisiones técnicas que no comprenden, se convertirán en blanco de las bromas del equipo de desarrollo, y la mayoría de las personas no pueden ser motivados por alguien a quien no respetan.
- *No involucrar a los desarrolladores en decisiones que les afectan.*- Puede dar la impresión de que el responsable no presta atención al equipo de desarrollo, o que ese directivo no lo respeta lo suficiente como para desear su participación. En la siguiente lista se muestran algunos ejemplos clásicos en los que la directiva debe comprometer a los desarrolladores si desea mantener alta la motivación:
 - Compromiso con los nuevos planes.
 - Compromiso con las nuevas prestaciones o modificaciones de las prestaciones del sistema.
 - Contratar a nuevos miembros del equipo.
 - Designar a desarrolladores para tareas a corto plazo fuera del proyecto.
 - Diseñar el producto.
 - Tomar decisiones de equilibrio técnico (por ejemplo, si para mejorar el rendimiento de la prestación A se debe disminuir el de B o viceversa).
 - Cambiar la oficina.
 - Comprometerse a entregar productos que podrían o no haber sido planeados por el equipo (por ejemplo, un prototipo que utilizará un cliente o una versión previa a la entrega del producto).
 - Comprometerse a nuevos procedimientos de desarrollo (por ejemplo, una nueva forma de control de cambio o una nueva clase de especificación de requerimientos).
- *Barreras para la productividad.*- Si el entorno se establece de forma que se frustra el esfuerzo del mejor desarrollador para ser productivo, se puede asegurar que la motivación se verá disminuida.
- *Baja calidad.*- Los desarrolladores derivan parte de su amor propio a partir de los productos que desarrollan. Si desarrollan un producto de gran calidad, se sienten bien. Por el contrario si desarrollan un producto de baja calidad, no se sienten bien. Para que un desarrollador se motive por orgullo de propiedad, se debe ser propietario del trabajo, y hay que estar orgulloso del trabajo. Pocos desarrolladores pueden sentir orgullo como respuesta al reto de desarrollar un producto de baja calidad en el menor tiempo posible.

Paro la mayoría de los desarrolladores están más motivados por la calidad que por la producción total. Si los directivos insisten en que los desarrolladores disminuyan la calidad para tener un menor tiempo de entrega, está reduciendo a mas de la mitad los motivadores de orgullo de propiedad.

- *Campañas cargantes de motivación.*- Con carteles, lemas, arengas y otros métodos de motivación es más fácil insultar la inteligencia del desarrollador que motivarlo.

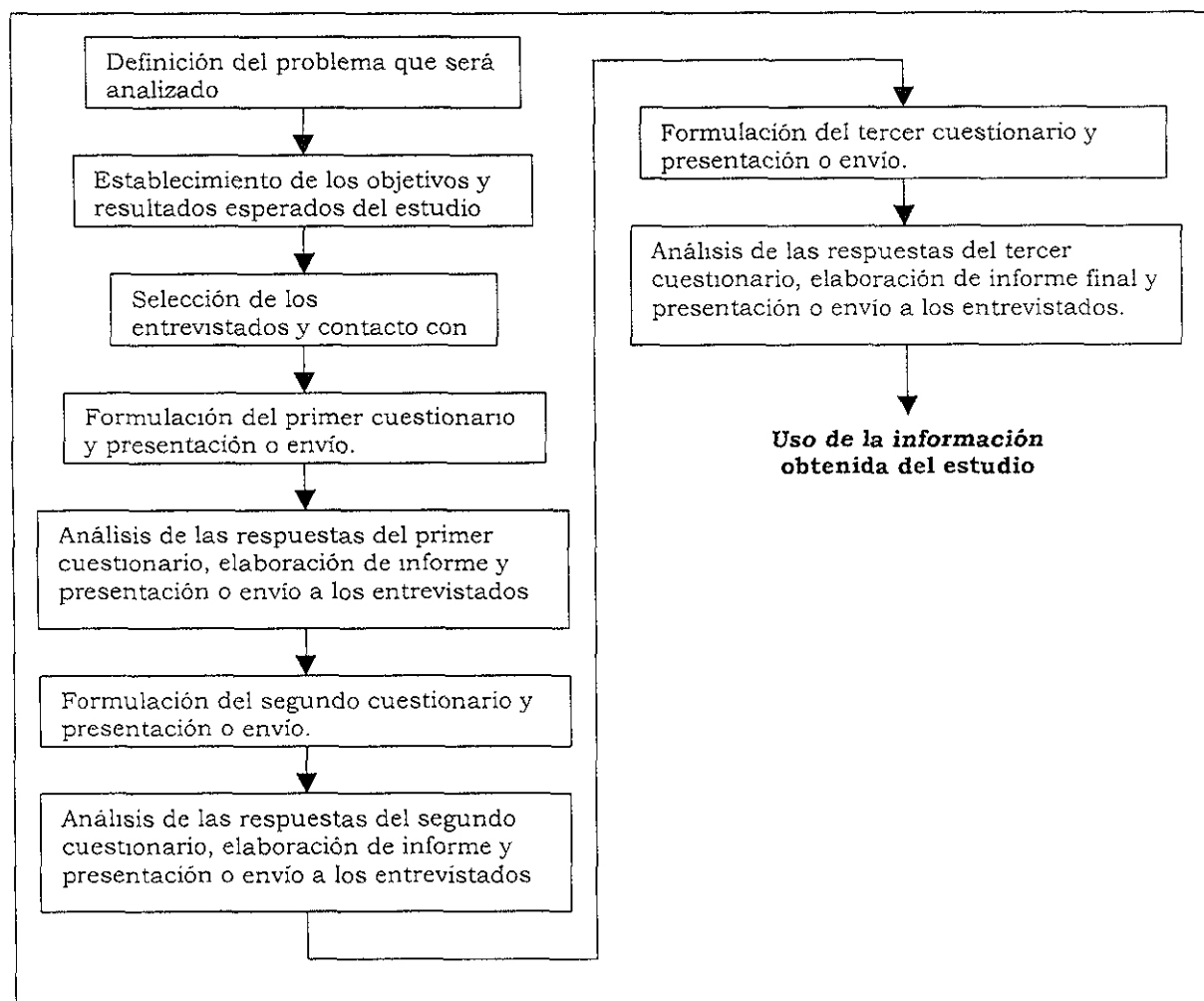
Finalmente resulta interesante la frase Frederick Herzberg para hacer conciencia sobre las mejores formas de motivar a los participantes en proyecto de desarrollo de software:

“Una patada en el trasero es una forma particularmente mala de motivación. Una patada en el trasero no produce motivación; sólo produce movimiento.”

Apéndice 3 Análisis de opiniones

Mucho del conocimiento que se aceptó como cierto a lo largo de este trabajo, se fundamentó en las opiniones que diferentes expertos en sistemas de información y en proyectos de este tipo, la forma en que se sintetizó se muestra a continuación.

El proceso de análisis de opiniones, por ejemplo para obtener los problemas frecuentes que se presentan en los proyectos, se ilustra en el siguiente diagrama:



Ejemplo de cuestionarios para la especificación de las diferentes etapas del proceso de administrar un proyecto de software.

Cuestionario referente al preanálisis

- ¿Cuáles son los datos mínimos con los que se puede realizar un análisis preliminar que permita estimar el esfuerzo necesario y el tiempo para realizar el proyecto que un cliente potencial requiere?
- ¿Es necesario agrupar los proyectos por tipos para facilitar este preanálisis?
- ¿Hay variaciones importantes entre diferentes tipos de proyectos sobre la información mínima indispensable para hacer un preanálisis?

Cuestionario referente a los criterios y lineamientos que se seguirán

- ¿Cómo se estima en Enlaces Tecnológicos la magnitud del esfuerzo necesario para realizar el proyecto?
- ¿Cuáles son los criterios que utilizan para asignar los proyectos a los diferentes participantes de la empresa?, ¿por disponibilidad de tiempo? o ¿por conocimiento en el tema?
- ¿Cuál es la estructura organizacional de los equipos de trabajo que ha manejado ET en sus proyectos?, ¿es un modelo único?, ¿existe un modelo más adecuado para algún tipo de proyecto que para otro?
- ¿Cómo hacen en la empresa para estimar el tiempo necesario para terminar un desarrollo?, ¿tienen políticas y lineamientos que unifiquen los criterios de la empresa?, ¿qué factores son relevantes para hacer la estimación según la experiencia de ET?
- ¿Con qué criterio se decide que tecnología es la más adecuada para realizar un desarrollo en específico (Con tecnología en este punto se hace referencia a las herramientas de programación que serán utilizadas)?
- ¿Cómo se decide cuál es el equipo (hardware) que deberá utilizarse en un desarrollo?
- ¿Cuál es la estructura de costos que utiliza la empresa?
- ¿Cómo definen el margen de utilidad? y ¿Qué aspectos son importantes en ET para hacer esta definición en proyecto determinado?

Cuestionario referente a la presentación de una propuesta

- ¿Existe un formato estándar que la empresa utiliza para presentar sus propuestas?, ¿está definido el tipo de papel, el tipo de letra y el orden que tendrá el contenido?
- ¿Qué puntos debe tener el contenido de la propuesta?

Cuestionario referente al análisis

- ¿Existen criterios definidos para la asignación del responsable de realizar el análisis?
- ¿Existen diferentes tipos de análisis que utilice la empresa?
- ¿Algún tipo de análisis aplica mejor para diferentes tipos de proyectos?, ¿en la empresa se debe utilizar un estándar para el análisis?
- ¿Existe un formato definido para presentar los análisis que ha realizado la empresa y que se puedan consultar?
- ¿Qué elementos debe contener un análisis?
- ¿Existe un procedimiento para verificar que un análisis esté bien hecho?

Cuestionario referente al diseño

- ¿Con qué criterios se selecciona al responsable de hacer el diseño?, ¿Por experiencia en proyectos similares?, ¿por conocimiento del tema?, o ¿por disponibilidad de tiempo? Entre otros
- ¿Existen diferentes tipos de diseños?
- ¿Algún tipo de diseño aplica mejor para diferentes tipos de proyectos?
- ¿Existen estándares definidos para realizar un diseño dentro de la empresa?
- ¿Cómo debe almacenarse la información de un diseño para que sea consultable posteriormente?

Cuestionario referente a la estimación

Tiempo

- ¿Cómo se calcula el tiempo en que un proyecto habrá de realizarse?
- ¿Qué parámetros debe tomarse en cuenta?
- ¿Con qué criterios se define el margen de seguridad?

- ¿Quién es el responsable de hacer el cálculo de este tiempo?
- ¿Alguien debe aprobar el cálculo?

Recursos humanos

- ¿Cómo se calcula el esfuerzo necesario para realizar un proyecto?
- ¿Qué factores deben ser tomados en cuenta?
- ¿Cómo se maneja un estándar de esfuerzo para diferentes actividades y que habrán de cumplir diferentes personas con capacidades diferentes?
- ¿Existe alguna unidad que la empresa utilice para medir el esfuerzo en proyectos de este tipo?

Recursos tecnológicos

- ¿Con qué criterios se decide qué equipo habrá de utilizarse en los proyectos?
- ¿En que se basan para decidir si se debe comprar más equipo?
- ¿Existen lineamientos para aprovechar lo mejor posible el equipo con que se cuenta?
- ¿Cómo se deciden por utilizar alguna herramienta de software en proyectos específicos?
- ¿Tienen planes de venta de equipo obsoleto y compra de nuevos?

Cuestionario referente a la definición de indicadores

- ¿Cómo se definen los entregables o puntos de control en los proyectos que ha participado la empresa?
- ¿Existen diferentes criterios para asignarlos?
- ¿Alguno de estos criterios es más adecuado para diferentes tipos de proyectos?
- ¿Una mala definición de los indicadores puede afectar en la entrega a tiempo del proyecto?

Cuestionario referente a la comunicación entre el equipo de trabajo y el cliente

- ¿Existen lineamientos de la empresa en los que se defina la forma en que se llevará la comunicación con el cliente y el representante de la empresa?
- ¿Con qué criterios se selecciona al responsable de representar a la empresa en un proyecto dado?
- ¿Qué facultades debe tener el representante?

Cuestionario referente a la gestión de riesgos

- Según la experiencia de la compañía ¿cuál es la probabilidad de que un proyecto se retrase?
- ¿Existen eventos comunes en los retrasos de diferentes proyectos?
- ¿Cuáles son?
- ¿Se contaba con planes de contingencia?
- ¿Cómo enfrentan los riesgos en los proyectos en que participan?

Cuestionario referente al plan de trabajo

- ¿Con qué criterios se organizan las actividades para ser asignadas a los participantes del proyecto?
- ¿Utilizan alguna herramienta que les ayude a hacer el programa de actividades?
- ¿Utilizan redes para representar el flujo de actividades?
- ¿Calculan la ruta crítica y las holguras de las actividades para hacer modificaciones en el plan?
- ¿Cómo definen el uso de los recursos económicos a lo largo del proyecto?

Cuestionario referente a la construcción

- ¿Existen en la empresa procedimientos estándar para la programación de los sistemas?
- ¿Con qué criterios deciden cuál será el lenguaje de programación?
- ¿Utilizan herramientas que faciliten el trabajo en grupo en sistemas grandes?
- ¿Cuáles son las reglas con las que presentan los informes de avance del proyecto?

Cuestionario referente al control

- ¿Cuáles son los instrumentos que utilizan para retroalimentar el plan de trabajo?
- ¿Cómo se cambia el plan cuando se presenta una desviación?
- ¿Quién es el encargado de informar al cliente el avance del proyecto?
- ¿Cómo se lleva la bitácora del proyecto para su almacenamiento y posterior consulta?

Cuestionario referente a las pruebas de calidad

- ¿Cómo se hacen las pruebas de modularidad, integración, de estrés y de interfaz con el usuario?
- ¿En qué momento es más apropiado realizarlas?
- ¿Con qué criterios se seleccionan los aspectos que incluirá una matriz de pruebas?
- ¿Existe alguna norma que sigan?
- ¿Levan algún registro de las pruebas que han efectuado?

Cuestionario referente a la entrega e instalación

- ¿Es común que sea necesario instalar los sistemas que desarrollan en diferentes unidades de operación?
- ¿Qué criterios utilizan para llevar a cabo la instalación de un sistema?
- ¿Utilizan algún modelo para optimizar la instalación?
- ¿Utilizan alguna herramienta para reducir los costos de instalación?
- ¿Cómo han capacitado a sus clientes para utilizar los sistemas que han realizado?

Cuestionario referente al plan de seguimiento posventa

- ¿Existe algún plan de seguimiento después de la entrega de un sistema para conocer su desempeño en el tiempo?
- ¿Ha sido común que después de haber realizado un sistema para un cliente surjan nuevas oportunidades de negocio?
- ¿Hacen un esfuerzo en la empresa por fomentar relaciones de largo plazo con sus clientes?
- ¿Cuáles son las políticas de Enlaces tecnológicos referentes a la garantía que ofrecen sobre sus productos?

Los cuestionarios que se presentaron en cada apartado de lo que debía definirse en un procedimiento de administración de proyectos informáticos sirvieron por una parte, para conocer con mas detalle la forma en que la empresa opera y la forma en que desea operar y el grado de madurez en relación con la administración de proyectos por la otra. Ayudó a fomentar la participación de los integrantes de la empresa en el diseño del procedimiento (con lo que se reduce el riesgo de rechazo cuando sea implantado) y permitió también crear una base de conceptos aceptados y entendidos que serán parte de la operación cotidiana en la empresa de ahora en adelante.

➤ Herramientas utilizadas

- Técnica TKJ
- Técnica DELFI
- Análisis TK
- Cuestionarios

Bibliografía

Steve McConnell

Desarrollo y gestión de proyectos informáticos

Microsoft Press

Mc Graw Hill 1997

Tom Gilb

Principles of software engineering management

Wokingham, England

Addison - Wesley 1988

Tom DeMarco

Controlling software projects

New York

Yourdon Press 1982

María Esther Espinosa Gómez

Consideraciones básicas para la implementación de la administración de proyectos de software (tesis)

DEPFI UNAM 1997

IT Professional (revista)

Artículos varios

IEEE Computer Society

1997 -1999

Henk W. Volberda

Building flexible organizations for fast-moving markets

Long range Planning, Vol. 30, No. 2, págs 169-183, abril 1997

Theodore Levitt

Comercialización creativa

CECSA 1986

Don Tapscott

Alex Lowy

David Tcoll

La era de los negocios electrónicos

Mc Graw Hill 1999

Charles H. Kepner

Benjamin B. Tregoe

El Nuevo directivo racional

Mc Graw Hill 1989

James A Senn
Análisis y diseño de sistemas de información 2ª Edición
Mc Graw Hill 1992

Alejandro Sierra Urrecho
Manuel Alfonseca Moreno
Programación en C/C++
Anaya multimedia 1999

Gabriel Sánchez Guerrero
Técnicas para el análisis de sistemas Parte I
2ª edición 1993
Departamento de Ingeniería de Sistemas,
División de Estudios de Posgrado,
Facultad de Ingeniería,
UNAM

Russell L. Ackoff
Un Concepto de Planeación de Empresas
Vigésima impresión
Ed. LIMUSA 1999

Kathy M. Ripin
Leonard R. Sayles
La contratación de servicios externos en sistemas de información
Ed. Oxford 2000