

Capítulo 4

Ciclo de vida de desarrollo de sistemas

Hasta hace poco tiempo los conceptos de planificación y gestión de proyectos se consideraron para los proyectos informáticos, aun en el desarrollo de los grandes sistemas se consideraban mas como una labor artesanal muy próxima al programador, que como una técnica con necesidad de una planificación efectiva.

Introducción

Actualmente el concepto de proyecto se aplica al campo de la informática debido a la evolución de los propios sistemas computacionales, la informática constantemente incrementa su capacidad y posibilidades, pero también las exigencias que debe cumplir, siendo la eficacia y rentabilidad de sus sistemas un factor muy importante para las organizaciones modernas.

Este factor importante ha traído como consecuencia el considerar la planificación y gestión de proyectos asociándole las técnicas y procedimientos para el desarrollo de sistemas computacionales o para cualquier otro tipo de proyecto, ya que del entorno al desarrollo y operación de sistemas se derivan infinidad de proyectos informáticos como se vio en el capítulo anterior.

El desarrollo de sistemas es ahora nuestro principal interés y se requiere que se utilicen los modelos y metodologías para el desarrollo, por lo que la gestión del proyecto debe normar la utilización de este tipo de técnicas y reconocerlas “como un conjunto organizado y documentado de procedimientos y guías de acción para una o más fases del ciclo de vida.”²⁴

El software, debe pasar por un conjunto de pasos para poder ser desarrollado, dichos pasos o etapas no están plenamente delimitados y se conocen como el *ciclo de vida de desarrollo de sistemas*, dado que cada desarrollo tiene su particularidad, en este capítulo se pretende mostrar los diferentes modelos de desarrollo desde un enfoque de gestión de proyectos.

²⁴ Gordon B. Davis Margrethe H. Olson, “Sistemas de Información gerencial”, Segunda Ed. McGraw Hill

4.1 Iniciación del proyecto

Para que pueda iniciar el ciclo de vida de desarrollo de sistemas, debe existir una solicitud de proyecto por parte del usuario, esta solicitud debe apegarse al procedimiento normativo de “Atención al usuario en el desarrollo de sistemas de cómputo”.²⁵

La gerencia del departamento usuario debe participar en la iniciación del proyecto y ésta debe evidenciarse en las actas administrativas, minutas de trabajo, programas de trabajo, presupuestos, calendarización, etc. de acuerdo al procedimiento establecido, no se debe iniciar un proyecto si el usuario no lo solicita, el imponer métodos de trabajo sin el deseo del usuario, generalmente fracasa.

			INSTITUTO MEXICANO DE INFORMATICA Subdirección General de Finanzas Jefatura de Servicios Informáticos		
SOLICITUD DE PROYECTO	Dependencia:		Fecha:		
	Usuario:		Folio:		
Dirección:		Ciudad:	Teléfono:		
DATOS DEL PROYECTO					
Clave:	Nombre:				
Peticiones del usuario:					
Objetivo:					
Alcance:					
Producto final esperado:					
Líder del Proyecto:					

Propuesta de formato para solicitud de proyecto

²⁵ Es indispensable que en las organizaciones exista una normatividad para las funciones de servicios informáticos.

4.2 Análisis y planificación

El objetivo de toda planeación es la de clarificar el problema a solucionar, definir el producto a obtener o servicio a proporcionar, estimar los costos económicos en que se va a incurrir, así como los recursos humanos y de cualquier otro tipo que se requieran para alcanzar la meta.

En la planeación se suelen distinguir dos grandes subfases:²⁶ la definición del problema y la definición del plan de desarrollo. Mientras que la primera se centra en clarificar el producto o servicio a obtener, la segunda define las actividades que se atenderán a lo largo del desarrollo del proyecto, anticipando los recursos que se requerirán, así como los momentos en que serán necesarios, además la secuencia y los tiempos en que se ejecutarán.

Es usual que los bienes o servicios solicitados tengan como condición el recurso escaso, por lo que el usuario debe seleccionar entre varias alternativas de solución. Así una mala definición de un proyecto puede ocasionar error en la toma de decisiones y que se comprometan los recursos de la organización en un bien no rentable.

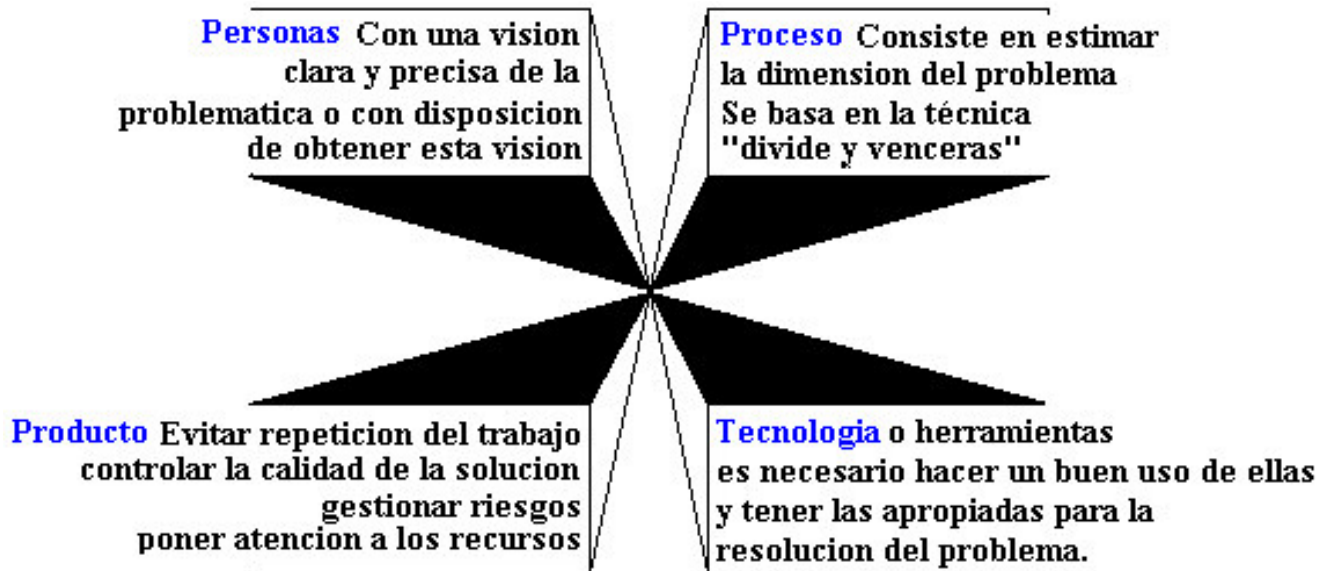
El origen de un proyecto suele ser difuso, normalmente alguien identifica un problema o una necesidad, este problema-necesidad puede verse desde diferentes enfoques, como un impedimento para alcanzar sus metas, mientras otros, lo ven como una oportunidad para dar una solución correcta.

Ya sea visto como problema u oportunidad, lo primero que hay que hacer es obtener una descripción clara de éste. La pregunta clave a responder es: *¿Cuál es el problema, o dónde está la oportunidad?* Evidentemente aquí hay que trabajar con los usuarios, directores de la organización y clientes, pues ellos son los que conocen su negocio y será de ellos de quien tendremos que obtener la información para responder a esta pregunta.

La definición del problema suele ocupar un cierto tiempo, por esto es muy importante darle la importancia central que tiene ya que todo el proyecto se basará en esta definición y es mejor que quede clara. La definición del problema debe ser revisada por todos los implicados en el problema. Para ello es conveniente tener en cuenta las cuatro dimensiones para la resolución de problemas²⁷:

²⁶ Wilson, Be. *Sistemas: conceptos, metodologías y aplicaciones*, Megabyte-Noriega editores, México, 1990.

²⁷ Shtub, A., Bard, J. Et Sholomo, G. *“Project Management Engineering, Technology and Implementation”*, Prentice-Hall, 1994.



Una de las fases mas complejas del proyecto es la de definir los objetivos, la persona que encarga el proyecto rara vez conoce claramente los objetivos, tan solo tiene una idea general, quiere informatizar algo o gestionar algo.

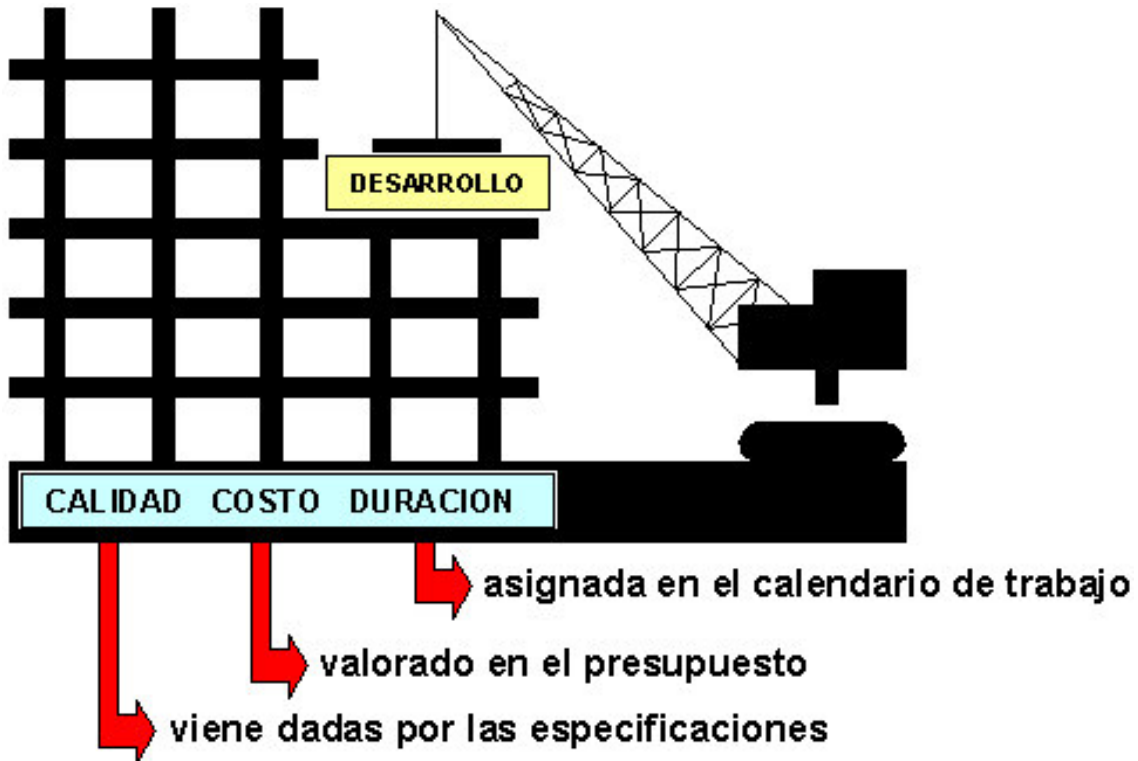
Este es uno de los problemas con que se encuentra el informático en las primeras fases del proyecto. El no definir los objetivos correctamente es la causa de muchos de los problemas que se presentan durante el ciclo de desarrollo del proyecto, los objetivos debe fijarlos pues quien encarga el proyecto y se ha de conseguir que estos sean claros, definidos, concretos y no ambiguos.

En función de la disponibilidad de los distintos recursos se evalúa la viabilidad del proyecto, es decir la garantía de acabarlo con éxito, así como el beneficio que reporta a la organización (Viable + Rentable = Proyecto iniciable).

Sin embargo la evaluación de la viabilidad es compleja en un proyecto informático, ya que a menudo no es posible estimar de forma correcta el costo (tiempo, trabajo, recursos, etc.) que va a conllevar una parte del proyecto.

Sobre la rentabilidad inciden directamente el tiempo de desarrollo y el tiempo de explotación, incluso si el proyecto va dirigido a un cambio en la infraestructura de la organización, como un nuevo programa gestor de contabilidad o a mejora de las comunicaciones, conlleva un costo para la organización, aunque al departamento solicitante le salga gratis.

La planificación del proyecto es la fase en la que se deberán identificar todas las cosas necesarias para poder alcanzar el objetivo marcado. En esta fase se han de concretar los tres cimientos sobre los que se apoyará el desarrollo de todo el proyecto, estos son:



En esta fase tendremos que identificar diferentes soluciones y los costos asociados a cada una de ellas. Las tareas a realizar para planificar el proyecto, las podemos agrupar en:

- Estimar el tamaño de la aplicación a desarrollar.
- Estimar el costo en recursos humanos.
- Identificar las tareas a realizar.
- Asignar recursos a cada tarea.
- Crear un calendario de las tareas.
- Realizar un estudio económico.
- Reunir todo en un documento, **Estudio de viabilidad.**

Todas estas tareas se suelen realizar de forma secuencial o iterando entre ellas, otro asunto es la secuencia a seguir.

Estudio de viabilidad²⁸

Todos los proyectos son realizables:

-idades recursos ilimitados y tiempo infinito!-

desafortunadamente, el desarrollo de un sistema se caracteriza por la escasez de recursos y límite de tiempo, esta premisa hace necesario y prudente evaluar la viabilidad de un proyecto.

Debe elaborarse un documento normativo que nos muestre la metodología para el desarrollo y documentación del estudio de viabilidad, lo anterior con el objeto de analizar los cursos alternos de acción que satisfagan los requerimientos de información del nuevo sistema.

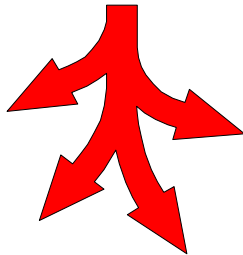
Los estudios de viabilidad deben contener:

Información general Se debe describir el origen y naturaleza de la solicitud del proyecto, su alcance.

Introducción Breve descripción del contenido del estudio.

Planteamiento del problema Se describirá al sistema actual, los problemas identificados, el objetivo general, las peticiones del usuario y los supuestos y restricciones.

Alternativas de solución Para cada alternativa de solución se deberá describir; el plan estimado de desarrollo, costos aproximados, beneficios, plan de operación, ventajas y desventajas.



Todas las alternativas propuestas atenderán las expectativas del usuario y variarán entre la mas modesta hasta la que considere tecnología de punta.

Recomendaciones Se anotará las conclusiones a las que se llegaron al analizar cada una de las alternativas de solución.

Cabe señalar que no se debe presuponer el recurso escaso, lo que se pretende es que el usuario conozca todas las posibles soluciones, una vez presentadas las alternativas de solución al grupo de control de calidad y a la gerencia usuaria, se seleccionará la alternativa que más convenga a la organización.

²⁸ Aguilar Castillo Gildardo, “Apuntes para la materia Administración de Recursos Informáticos”, Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Se deberá desarrollar un estudio de factibilidad para la alternativa seleccionada si ésta contempla la adquisición de hardware, software o algún recurso que implique gasto ó inversión.

Estudio tecnológico de factibilidad²⁹

Debe prepararse y documentarse un estudio tecnológico para la alternativa seleccionada en el estudio de viabilidad y debe contener la siguiente información:

- Necesidades de hardware y software y su disponibilidad,
- Validar restricciones de espacio y tiempo implícitas en los requerimientos de información del departamento usuario y la manera de satisfacerlas.
- Factibilidad operacional, de cómo el nuevo proyecto encaja en la actual mezcla de hardware, software.
- Consideraciones legales.

Debe existir una metodología de trabajo para el desarrollo de estudios de factibilidad plasmada en un documento normativo que nos muestre paso a paso las actividades que se deben realizar y que este de acuerdo a la normatividad de la organización, obviamente las actividades variaran dependiendo de la naturaleza del proyecto, a continuación algunas actividades típicas:



- Posibilidades de outsourcing.
 - Análisis de las aplicaciones de la organización.
- Crecimiento de la organización.
- Prediseño de los sistemas actuales considerando el impacto.
 - Análisis de los requerimientos de información.
 - Análisis de los requerimientos de hardware y software.
 - Requerimiento de espacios físicos e instalaciones.
 - Análisis de los proveedores y sus propuestas.
- Forma de adquisición de equipo.
 - Análisis del rendimiento del hardware y software.
 - Costos.
- Comprobación de propuestas.
 - Selección de la propuesta.

²⁹ Idem.

4. Ciclo de vida de desarrollo de sistemas

El estudio de factibilidad técnica podrá hacerse de la siguiente forma:

1. Una breve descripción de la situación actual en la cual se reflejen los puntos más importantes, hallazgos y recomendaciones dirigidos a la alta gerencia.
2. Una descripción detallada de las especificaciones funcionales del equipo para cada alternativa.



- Determinar el rendimiento operativo del equipo a través de "benchmarks" ³⁰ y otras .
 - Pruebas de campo para cada alternativa.
 - Determinar la capacidad, expandibilidad y modularidad de los equipos en el mismo medio de trabajo o en condiciones similares.
 - Evaluar la calidad y facilidades del mantenimiento, soporte, entrenamiento y documentación del software y hardware.
 - Considerar la reputación del vendedor, experiencia, capacidad de actualizar el Hardware y software a menor costo.
 - Considerar en su totalidad los costos de cada alternativa.
4. Se debe señalar claramente la propuesta técnicamente mas recomendable, sus fortalezas y debilidades así como fechas compromiso.
 5. Se debe romper la resistencia a la lectura que tienen algunos ejecutivos por medio de conclusiones concretas que sean sencillas y su presentación debe ser preparada utilizando técnicas audiovisuales.

En términos generales se deben analizar:

- costos del sistema operacional
- ahorros y beneficios para la organización
- impacto en la organización y procedimientos actuales

La gerencia debe revisar los reportes de estudios de factibilidad y decidir si procede; si la decisión es continuar debe seleccionar alguna de las propuestas, paso siguiente se debe desarrollarse un plan maestro.

³⁰ Aplicaciones usadas para medir el rendimiento de una computadora.

Planificación del proyecto ³¹

Debe desarrollarse un plan maestro que incluya los procedimientos adecuados para mantener el control sobre el proyecto que se está desarrollando, dividir el proyecto en etapas y actividades que se puedan cuantificar. El plan maestro debe incluir un método para controlar los costos durante las diversas fases del ciclo de vida de desarrollo del sistema.

Planificación del proyecto es el curso de acción a seguir considerando actividades, tiempos y costos necesarios para realizar un trabajo determinado.

Objetivos:

- * Contar con un registro de todas las actividades que se tienen que desarrollar.
- * Proporcionar al equipo del proyecto una guía que les permita observar las actividades a realizar y los tiempos a que se deben ajustar.
- * Disponer de información que permita conocer el avance o retraso de las actividades definidas dentro del plan.
- * Proporcionar al responsable del proyecto información que le permita planear actividades posteriores o diferentes a las descritas dentro del plan o disponer de los recursos que hayan terminado sus actividades asignadas.



Consideraciones para el diseño del plan:

- * Definir las políticas de trabajo.
- * Determinar las actividades a desarrollar.
 - * Ordenar en forma lógica dichas actividades.
 - * Señalar la importancia de cada actividad.
- * Determinar los recursos necesarios para cada actividad.
- * Asignar tiempos en general y para cada una de las actividades.

³¹ Idem.

4. Ciclo de vida de desarrollo de sistemas

Técnicas de planeación: Gráficas de Gantt

Consiste en anotar en forma de lista las actividades a realizar en un orden lógico, señala el tiempo de duración de cada actividad por medio de una barra que abarca los períodos que necesita para cumplirse, en dónde se termine una actividad se debe empezar la siguiente, de tal forma que se vaya anotando el tiempo que dura cada actividad y el tiempo que se va acumulando, se deben presentar dos columnas que muestren el tiempo total estimado y el tiempo real que se utilizó.

Ventajas

- * Sencillez
- * Muestran el tiempo real y estimado
- * Es una base para el control
- * No requiere de especialistas para su elaboración.

Desventajas

- * No considera la importancia de algunas actividades.
- * No indica muy claramente el objetivo.
- * No se muestra muy bien la relación, de las actividades entre sí.

Ejemplo de un formato utilizando esta técnica.

Actividad		R	FECHA		Graficación: Día - mes													
			Inic	Term	Ene	Feb	Mar	Abr										
1	Planteamiento de los requerimientos	1	Ene 2	Ene 12	█													
2	Análisis de la información documental	2	feb 4	feb 19		█												
3	Entrevista al usuario	1 2	feb 20	may 5			█											
4	Determinación de los productos esperados	2	may 10	may 30				█										

VoBo: Jefe Dpto. Informática
Ing. Gildardo Aguilar Castillo

Día x Semana Mes

Técnicas de planeación: Redes

Consiste en ordenar las actividades necesarias, en forma sucesiva y lógica e interrelacionadas de manera tal, que conduzcan a una meta predeterminada con el menor desperdicio de tiempo, debe representarse por medio de una malla o red. Permite visualizar el camino crítico o sea la serie de actividades que no deben retrasarse y que en suma constituyen la duración del trabajo, se visualizan también las actividades no críticas, esto es con cierto margen de tolerancia, también las actividades que se pueden ejecutar en forma simultánea para optimizar el tiempo. Representa la relación y dependencia de todas las actividades entre sí.


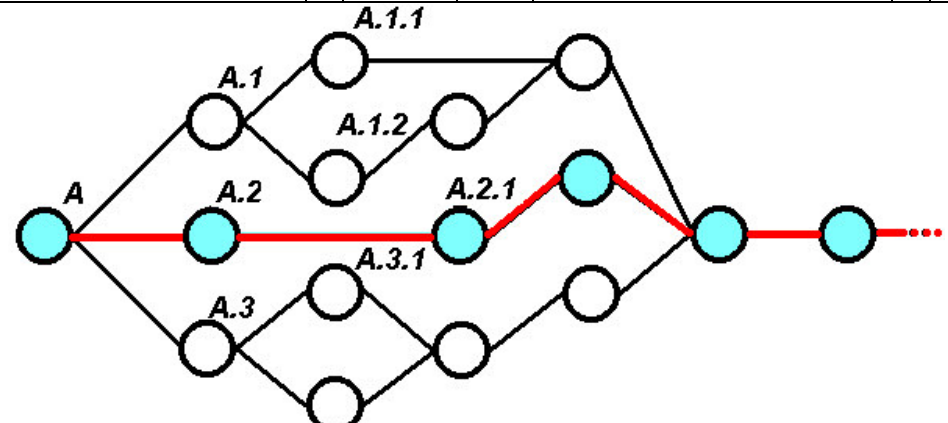
Ventajas

- * Muestra claramente el objetivo.
- * Permite visualizar la relación entre las actividades.
- * Señala las actividades críticas o sea las más importantes.
- * Ayuda a vigilar las actividades importantes.

Desventajas

- * Requiere análisis.
- * Necesita de especialistas para elaborarse.
- * Complejidad relativa.

Ejemplo de un formato utilizando esta técnica.

	<p>INSTITUTO MEXICANO DE INFORMATICA Subdirección General de Finanzas</p> <p>Jefatura de Servicios Informáticos</p>																																
Nombre del proyecto: <i>Equipamiento al Departamento de Personal</i> Clave deL Proyecto: <i>IP-012/03</i>																																	
RESPONSABLES																																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; text-align: center;">1</td> <td style="text-align: center;"><i>Ing. David Medrano Félix</i></td> <td style="width: 50%; text-align: center;">3</td> <td style="text-align: center;"><i>CP. Roberto Gómez Junco</i></td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;"><i>LI. Emilio Fernando Alonso</i></td> <td style="text-align: center;">4</td> <td></td> </tr> </table>	1	<i>Ing. David Medrano Félix</i>	3	<i>CP. Roberto Gómez Junco</i>	2	<i>LI. Emilio Fernando Alonso</i>	4																										
1	<i>Ing. David Medrano Félix</i>	3	<i>CP. Roberto Gómez Junco</i>																														
2	<i>LI. Emilio Fernando Alonso</i>	4																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No</th> <th>Actividad</th> <th>R</th> <th>Fecha</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	No	Actividad	R	Fecha													<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>No.</th> <th>Actividad</th> <th>R</th> <th>Fecha</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table>	No.	Actividad	R	Fecha												
No	Actividad	R	Fecha																														
No.	Actividad	R	Fecha																														
																																	

4.3 Diseño y desarrollo

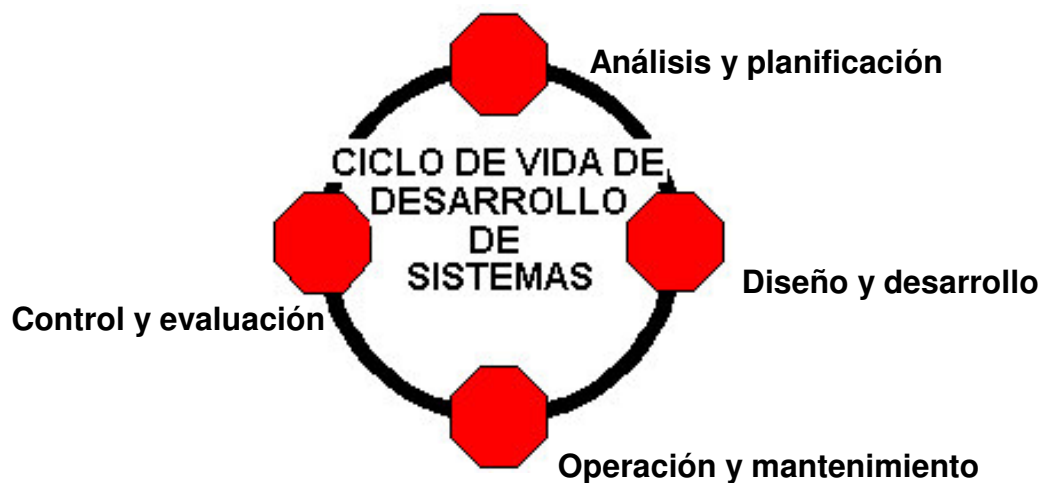
Diseño de un proyecto

El desarrollo de software tiene su principal justificación en la respuesta que se ha de dar al sistema de información que se encuentra funcionando en la organización. Existen dos tipos de sistemas de información: ³² Los sistemas de información formales que se distinguen por la estructura de sus procedimientos, sin entrar en detalle que tan óptimos son, y los sistemas de información informales que utilizan acuerdos implícitos o reglas no establecidas de comportamiento.

Evidentemente que los desarrollos de sistemas computacionales deben tener en cuenta una solución organizacional y administrativa basada en las tecnologías de la información para afrontar los retos impuestos por el entorno, por lo que involucra más que el uso de computadoras, del entendimiento de la organización, la administración y la tecnología, debido a ello es que ahora se habla de proyectos informáticos y no sólo de sistemas.

Para poder trasladar las especificaciones funcionales que se definieron en la alternativa seleccionada de la etapa de análisis es necesario utilizar una serie de modelos y metodologías para la determinación y el diseño de los procesos y datos que se requieren, además del aprovechamiento óptimo de los recursos de hardware y software,

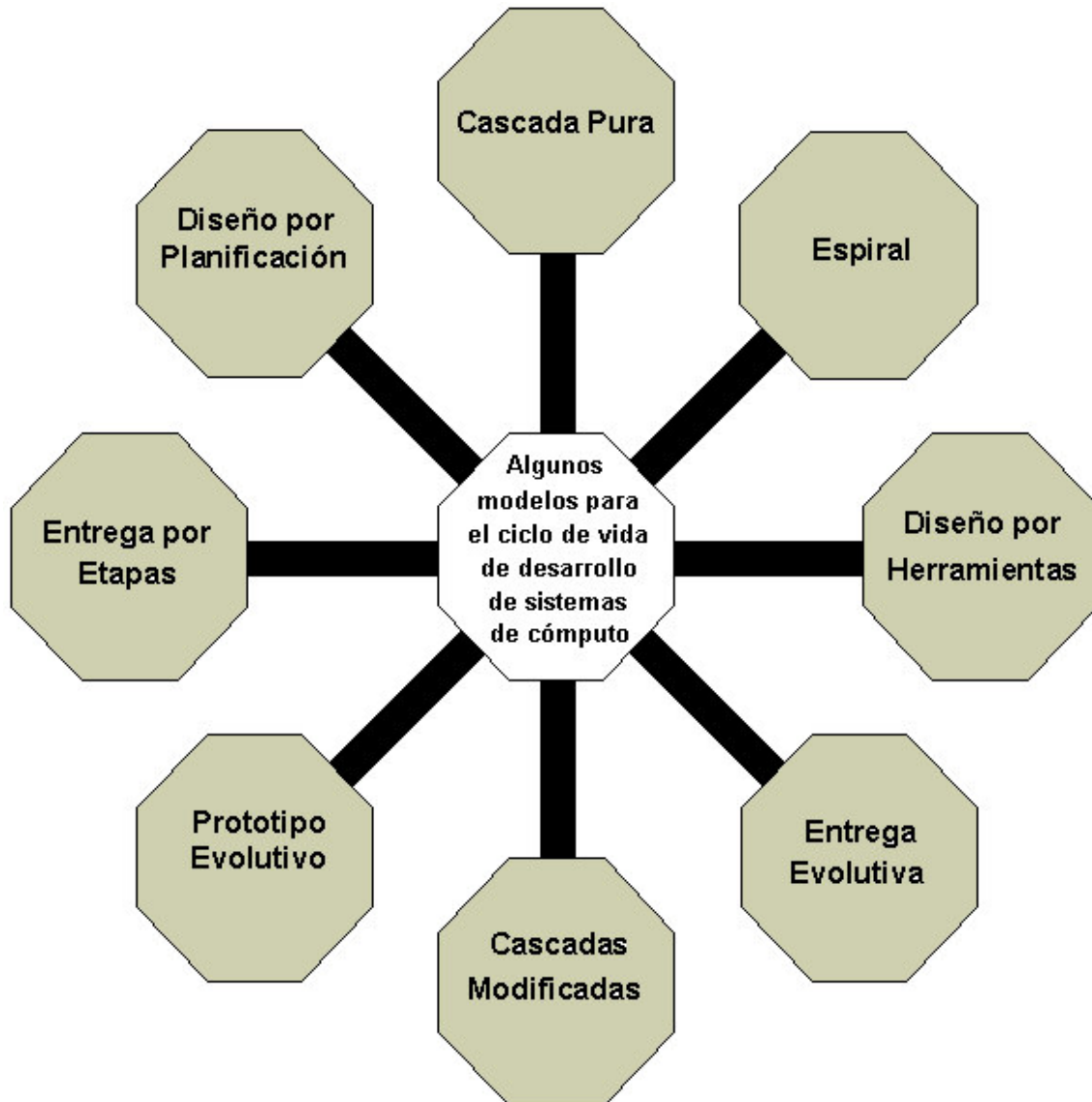
Todo esfuerzo en el desarrollo de un sistema computacional conlleva un ciclo de vida, el cual es un modelo prescriptivo de lo que pasaría entre la primera idea y el funcionamiento del sistema. Existen varios modelos del ciclo de vida, elegir el apropiado ayuda a orientar el proyecto y a asegurar que cada paso se acerque más a la consecución del objetivo. El ciclo de vida tiene un patrón circular.³³



³² Bird, Hair, Lamb, *Marketing*, International Thompson Editores, 7a. Ed., 2000

³³ Aguilar Castillo Gildardo, "Apuntes para la materia Administración de Recursos Informáticos", Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Dependiendo del modelo de ciclo de vida seleccionado se puede aumentar la velocidad del desarrollo, mejorar la calidad, el control y el seguimiento del proyecto, minimizar gastos y riesgos, mejorar las relaciones con el usuario. Por ello, la mala selección de un modelo de ciclo de vida hace más lento, repetitivo, innecesario y frustrante el trabajo.



Algunos modelos para el ciclo de vida³⁴ de desarrollo de sistemas

³⁴ Checkland, Peter and Shoges, Jim. *Metodología de los sistemas suaves de acción*, Megabyte – Noriega editores, México.

Cascada pura

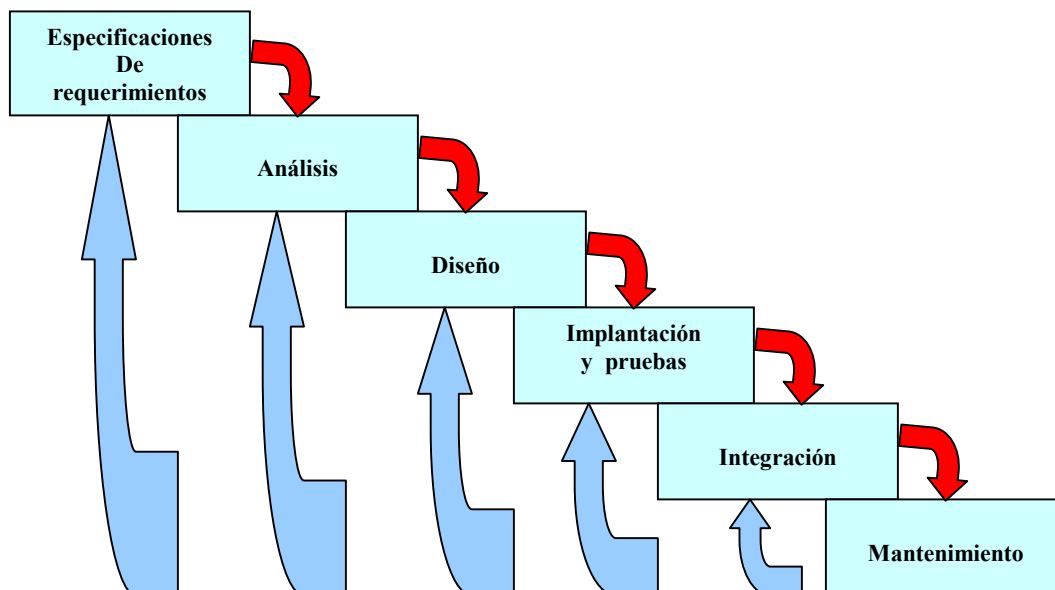
Es el antecesor de todos los modelos de ciclo de vida y ha servido de base para varios de ellos. En este modelo el proyecto progresa a través de una secuencia ordenada de etapas, partiendo desde su concepto inicial hasta la prueba del mismo. Se realiza una revisión al final de cada etapa del proyecto, para determinar si se está preparado para pasar a la siguiente.

Ventajas:

- Se utiliza para ciclos en los que se tiene una definición estable del proyecto, o cuando se está trabajando con metodologías y técnicas conocidas.
- Puede constituir una elección correcta para el desarrollo rápido cuando se está construyendo una versión de mantenimiento bien definida de un producto existente, o cuando se está migrando un producto existente a una nueva plataforma.
- Ayuda a minimizar los gastos de planificación.
- Evita una fuente común de errores importantes eliminando los cambios que se puedan producir a medio camino.
- Su estructura ayuda a minimizar el esfuerzo inútil.

Desventajas:

- Resulta muy difícil volver atrás utilizando el modelo de cascada pura.
- Genera pocos signos visibles de progreso hasta el final.
- Es poco flexible.



Una típica representación de desarrollo de software en cascada³⁵

³⁵ Aguilar Castillo Gildardo, “Apuntes para la materia Administración de Recursos Informáticos”, Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Espiral

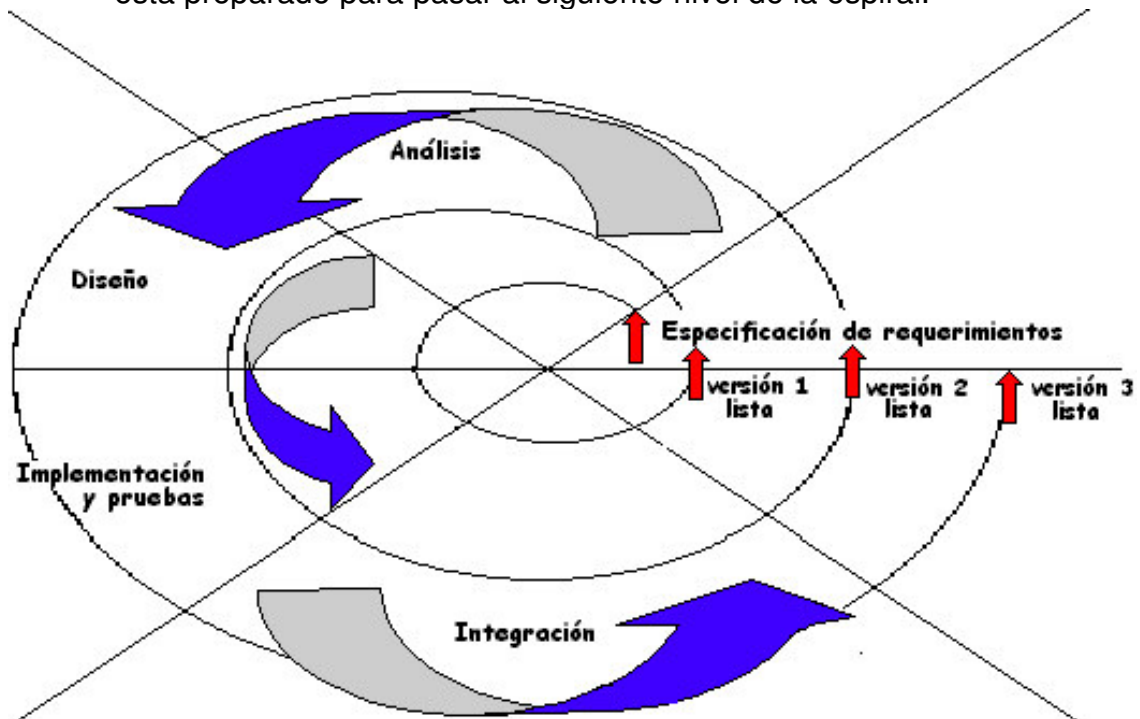
Este modelo está orientado a riesgos, por lo que divide el proyecto en miniproyectos, cada uno de los cuales se centra en uno o más riesgos hasta que todos están controlados. Después de controlar todos los riesgos el modelo finaliza del mismo modo que el modelo de ciclo de vida en cascada.

Ventajas:

- Aunque sube los costos disminuye los riesgos.
- Proporciona al menos tanto control de gestión como el modelo de cascada tradicional.
- Como el modelo está orientado a riesgos, proporciona con anterioridad indicaciones de cualquier riesgo insuperable.
- Es posible descubrir si el proyecto no se puede realizar por razones técnicas u otras razones.

Desventajas:

- Es un modelo complicado, requiere una gestión profunda y atenta.
- Puede ser difícil definir objetivos de comprobación que indiquen si esta preparado para pasar al siguiente nivel de la espiral.

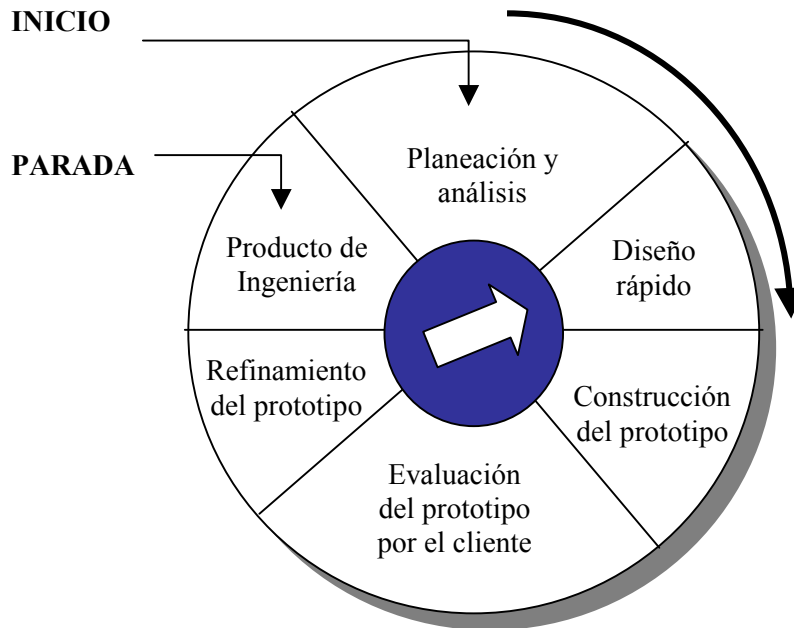


Una representación de desarrollo de software en espiral³⁶

³⁶ Aguilar Castillo Gildardo, “Apuntes para la materia Administración de Recursos Informáticos”, Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Prototipado evolutivo

En este modelo de ciclo de vida se desarrolla el concepto del sistema a medida que avanza el proyecto. Se inicia desarrollando los aspectos más visibles del sistema. Se presenta al cliente la parte ya desarrollada del proyecto y se continúa el desarrollo del prototipo con base en la realimentación que se recibe del cliente. El ciclo continúa hasta que el prototipo se convierte en el producto final de ingeniería. La gráfica del presente modelo sería la siguiente:



Una representación de desarrollo de software en prototipado evolutivo³⁸

Ventajas:

- Ideal para proyectos cuyos requerimientos cambian con rapidez.
- Cuando el cliente no puede especificar el conjunto total de los requerimientos.
- Cuando no se logra identificar de forma apropiada el área de aplicación.
- Cuando los desarrolladores no están seguros de la arquitectura o los algoritmos adecuados a utilizar.

Desventajas:

- Existe una imposibilidad de conocer al inicio del proyecto lo que se tardará en crear un producto aceptable.
- Esta aproximación puede convertirse fácilmente en una excusa para realizar el desarrollo con el modelo de codificar y corregir.

³⁸ Zachman, John A. *El modelado de las empresas: la arquitectura de Zachman*. Zachman Institute for Framework Advancement. Estados Unidos, 1999.

Entrega por etapas

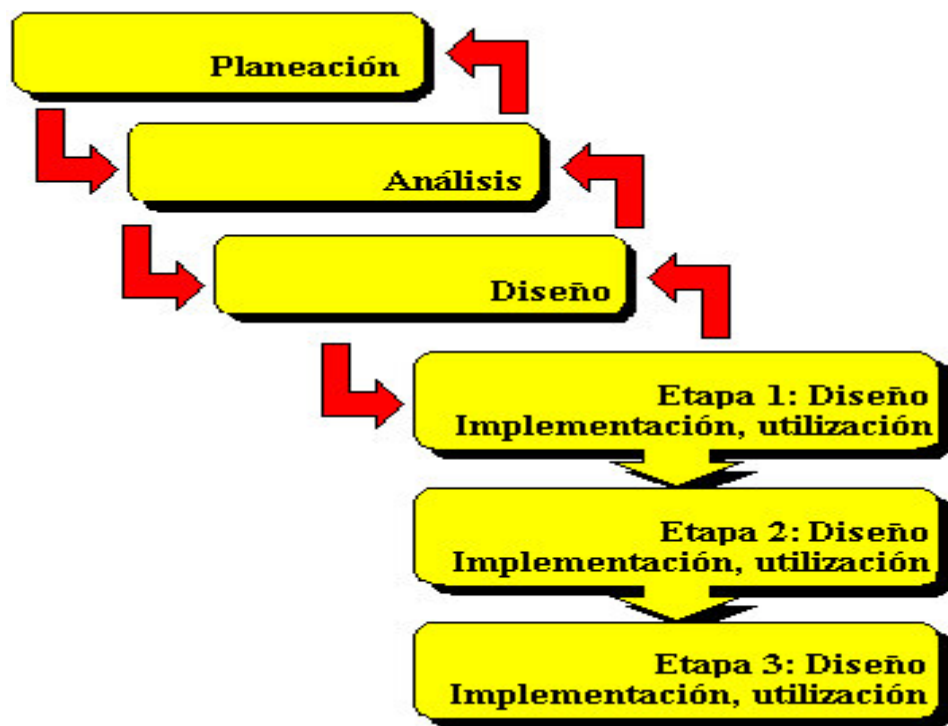
En este modelo de entrega por etapas, el sistema se muestra al cliente en etapas refinadas sucesivamente, conociéndose con exactitud lo que se va a construir desde el principio. En este modelo no se entrega el sistema como un todo al final del proyecto, sino que se entrega por etapas sucesivas a lo largo del proyecto.

Ventajas:

- Permite proporcionar una funcionalidad útil en las manos del cliente antes de entregar el 100% del proyecto.
- Con una planificación cuidadosa, es posible entregar las prestaciones más importantes al principio, y el cliente puede comenzar a usar el sistema en ese punto.
- Proporciona signos tangibles de progreso en el proyecto.

Desventaja:

- No funciona sin una planificación adecuada.



Una representación de desarrollo de software en entrega por etapas³⁹

³⁹ Zachman, John A. *El modelado de las empresas: la arquitectura de Zachman*. Zachman Institute for Framework Advancement. Estados Unidos, 1999.

Diseño por planificación

Es similar al modelo de entrega por etapas, pero se diferencia en que no siempre se conoce al principio si se tendrá el producto para la última entrega. Se pueden tener cinco etapas planificadas, pero sólo se llega a la tercera etapa debido a que se tiene una fecha límite que no se puede cambiar.

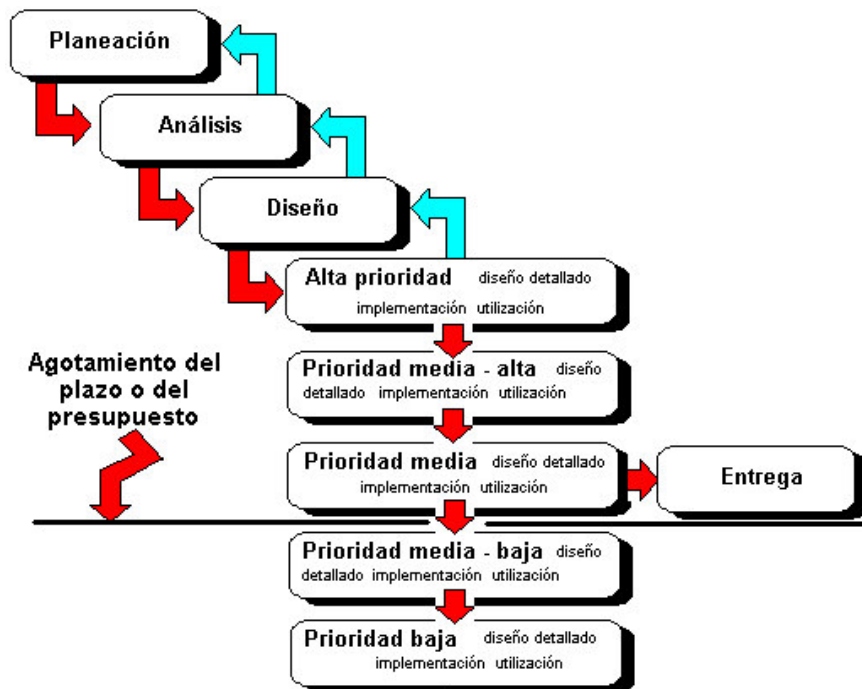
Uno de los elementos críticos de este modelo es priorizar los requerimientos y planificar sus etapas de tal manera que las primeras contengan los requerimientos de mayor prioridad, los requerimientos de baja prioridad se dejan para más tarde.

Ventajas:

- Puede ser una estrategia válida para asegurar que se tiene un producto listo en una fecha determinada.
- Esta estrategia es particularmente útil para las partes del producto que no se quieren realizar en el camino crítico.

Desventajas:

- Si no se completan todas las etapas, se desperdiciará tiempo en la especificación, arquitectura y diseños de prestaciones que no se van a entregar.
- Si se ha gastado tiempo en una gran cantidad de requerimientos incompletos que no se van a entregar, se debería tener tiempo para resumir en uno o dos requerimientos más completos.



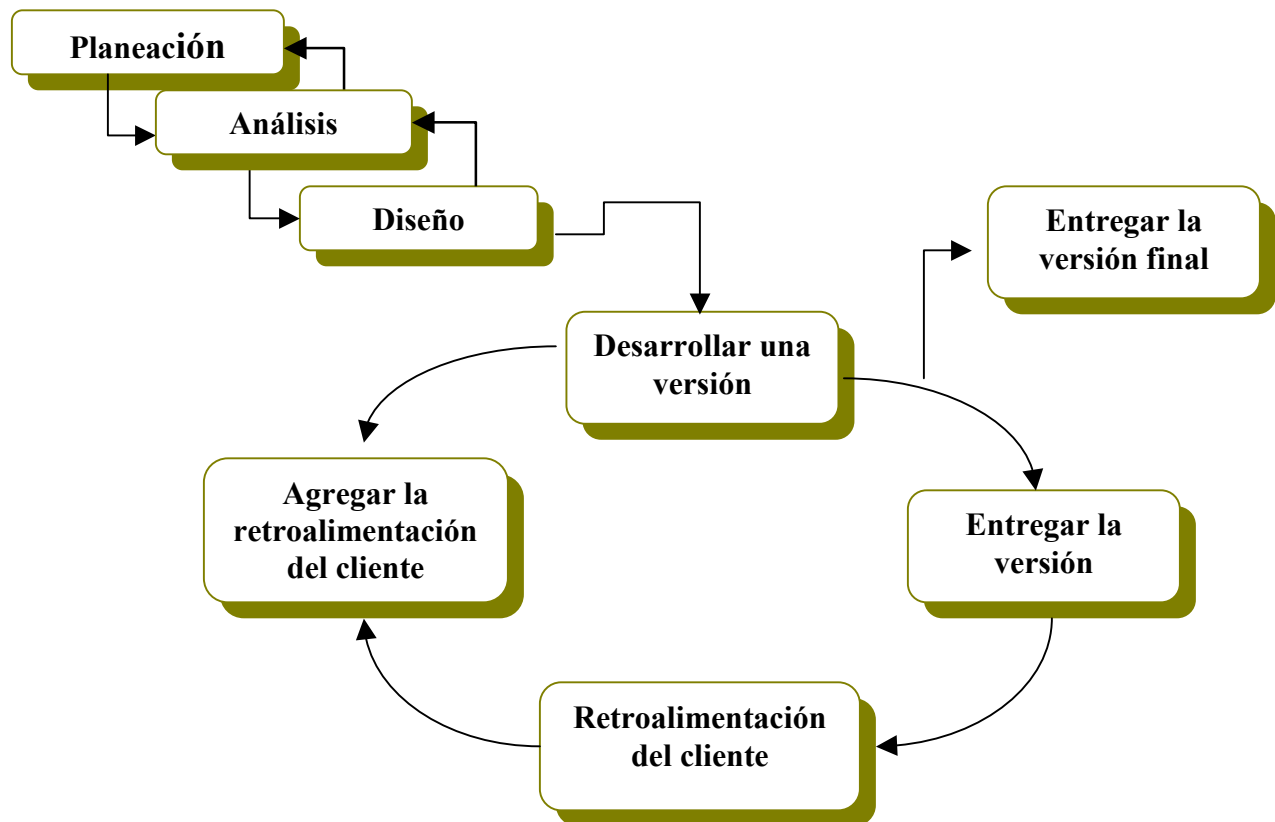
Una representación de desarrollo de software Diseño por planificación ⁴⁰

⁴⁰ Aguilar Castillo Gildardo, “Apuntes para la materia Administración de Recursos Informáticos”, Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Entrega evolutiva

Es un modelo que se encuentra entre el prototipado evolutivo y la entrega por etapas, puesto que se desarrolla una versión del producto, se muestra al cliente, se refina el producto en función de los comentarios del cliente. El parecido entre ambos modelos depende de hasta qué punto se lleva a cabo una planificación para adaptarse a las solicitudes de los clientes.

Si se planifica para adaptarse a la mayoría de las solicitudes, la entrega evolutiva se parecerá más al prototipado evolutivo; en cambio si se planifica para adaptarse a pocas solicitudes de modificación se parecerá más a la entrega por etapas.



Una representación de desarrollo de software entrega evolutiva ⁴¹

⁴¹ Zachman, John A. *El modelado de las empresas: la arquitectura de Zachman*. Zachman Institute for Framework Advancement. Estados Unidos, 1999.

Diseño por herramientas⁴²

La idea es incluir una funcionalidad dentro del producto sólo si las herramientas de software existentes la soportan directamente. Ejemplos de herramientas son librerías de código y clases, generadores de código, lenguajes de desarrollo rápido y otras herramientas software que reducen de manera espectacular el tiempo de implementación.

Ventaja:

- Se puede combinar con otros modelos.

Desventajas:

- Se pierde mucho control sobre el producto.
- Puede que no sea posible llevar a cabo la implementación de todos los requerimientos que se desean, y que no se puedan implementar otros requerimientos exactamente de la forma que se quiere.
- Depende en buena medida de los productores de software comercial.

Codificar y corregir⁴³

Es un modelo poco útil pero muy común. Cuando al realizar un proyecto no se elige ningún modelo en específico, por omisión se está utilizando éste. Cuando se utiliza se inicia con una idea general de lo que se necesita construir y se continúa así hasta terminar el proyecto.

Ventajas:

- No conlleva ninguna gestión.
- No se pierde tiempo en la planificación, la documentación, el control de calidad, o cualquier otra actividad que no sea la codificación pura.
- Muestra inmediatamente indicios de progresos.
- Requiere poca experiencia.

Desventajas:

- Resulta peligroso para proyectos grandes.
- No ofrece medios de evaluación del progreso, ni de calidad o identificación de riesgos.
- No se cuenta con antecedentes documentales.

Consideraciones en la etapa

Durante las fases de diseño deben establecerse la entrada, la salida, los archivos y las especificaciones de procesamiento y deben diseñarse los documentos fuente, los controles y las pistas de Auditoría.

⁴² Checkland, Peter and Shoges, Jim. *Metodología de los sistemas suaves de acción*, Megabyte – Noriega editores, México.

⁴³ Idem

4. Ciclo de vida de desarrollo de sistemas

Algunas consideraciones que debe tener en cuenta el equipo del proyecto en la fase de Diseño de Sistemas.

1. Basarse en la planificación del proyecto.
2. Todo diseño de sistemas debe tener como base la documentación correspondiente a la fase del análisis del sistema.
3. Definir controles de carácter manual, mecanizado y operacional.
4. Evaluar el grado de exactitud de todos y cada uno de los programas que forman parte del sistema.
5. Diseñar los documentos fuente necesarios.
6. Determinar la edición y validación de los datos.
7. Seleccionar el tipo de organización de archivos, base de datos.
8. Definir los modos de procesamiento en relación al estudio de viabilidad.
9. Diseñar los productos y salidas.
10. Diseñar los procedimientos de seguridad del sistema.
11. Elaborar la diferente documentación del sistema.

Fases de desarrollo

En esta fase inicia tras la aprobación formal del diseño ⁴⁴, se realizan básicamente tres funciones, la programación-prueba de los programas, la elaboración de los manuales (operación, usuario..) y la implantación.

Para las actividades de programación es recomendable integrarlas en dos fases, primeramente el diseño de alto nivel o arquitectónico y posteriormente el diseño detallado, el primero tiene como objetivo definir la estructura del software a partir del modelo establecido en la fase anterior, se obtiene asignando funciones a componentes software, definiendo los flujos de datos e implementando controles.

Algunas consideraciones⁴⁵ que debe tener en cuenta en el diseño de alto nivel:

1. Se debe adoptar y utilizar una metodología de diseño de software concreta y reconocida.
2. La arquitectura del software se descompondrá en unidades de menor complejidad, preferentemente se adoptara una metodología de tipo “top-down”.
3. Para cada elemento o componente se indicará al menos los datos de entrada, las funciones que desempeña y los datos de salida.

⁴⁴ Es muy recomendable que como actividad final en el diseño, se presente a un comité de calidad, formado por el personal mas calificado de la organización en materia de informática.

⁴⁵ Domingo Ajenjo Alberto, “Dirección y Gestión de Proyectos, un enfoque practico “ Ed. Alfaomega, Madrid España, 2000, pp 294.

4. Se estimarán durante esta fase los requerimientos computacionales para el entorno de desarrollo y operativo.
5. Se elaborará la documentación correspondiente que muestre la correspondencia entre los requisitos del software y los elementos del diseño.

En el diseño detallado, se refinan los detalles mas significativos del diseño de alto nivel, se codifica, documenta y prueba. Algunas consideraciones que debe tener en cuenta el equipo del proyecto con las actividades de programación y prueba de los programas.

1. Todo programa realizado debe apegarse estrictamente a las especificaciones funcionales (carpeta del programa)⁴⁶ y del diseño de alto nivel.
2. Se deben aplicar las técnicas de programación concretas y reconocidas, así como realizar pruebas de escritorio.
3. Debe existir un plan de prueba para asegurar que funcione el programa en todas sus opciones y ante diferentes situaciones.
4. Los resultados de las pruebas deben ser consistentes con las especificaciones originales.
5. En plan para la prueba integral del sistema debe intervenir el departamento usuario para definir los estándares de prueba de sistemas.
6. Las pruebas deben considerar el tamaño de la instalación, la complejidad de procesamiento y la naturaleza de los datos.
7. Se debe incluir la bitácora de los tiempos de procesos y las actas o minutas de aceptación del sistema.

Algunas consideraciones que debe tener en cuenta el equipo del proyecto con las actividades de la elaboración de los manuales.

Manual de operación.- Es un manual que documenta las especificaciones de procesamiento de los sistemas, son dirigidos a los operadores de equipos que administran los procesos de varias aplicaciones, deben ser preparados y documentados para probarlos junto con el sistema, deben ser individuales por sistema y estar de acuerdo a los estándares de documentación, usualmente deben contener la información necesaria para el operador sobre cada parte de la cadena de procesos, para cada paso del trabajo la documentación deberá especificar:

- Requerimientos de procesamiento.
- Control de proceso a proceso.
- Explicación de todos los mensajes de consola y su respuesta adecuada.

⁴⁶ Algunas áreas de desarrollo, incluyen en su normatividad a la carpeta de programa, que incluye un objetivo, un narrativo, las entradas utilizada, las salidas producidas, los archivos y/o bases de datos.

4. Ciclo de vida de desarrollo de sistemas

- Procedimientos específicos para notificar errores o condiciones de falla.
- Puntos apropiados de reinicio.
- Creación de salidas y su disposición.
- Identificación adecuada de las etiquetas de archivos.

Manual de usuario.- Es un documento sencillo dirigido a los usuarios cuando éstos operan directamente su sistemas en procesos distribuidos o se tienen procesos interactivos en la aplicación, debe contener la misma información que el manual de operación solo que redactada de forma mas amigable para que la puedan entender, la redacción es la única diferencia. En algunas organizaciones además de los puntos anteriores incluyen temas como:

- Asignación de prioridades y tiempos de respuesta.
- El calendario y la frecuencia de la distribución de los producto.
- La seguridad, vigencia y disposición de las salidas.
- Procedimiento para encender y apagar estaciones de trabajo.
- Solución a fallas de hardware de primer nivel
- Procedimiento para firmas de entrada.

Los manuales de operación y del usuario deben estar elaborados antes de la prueba final del sistema, para probarse de igual forma y deben ser distribuidos de acuerdo a las políticas de la organización.

Implantación

Cuando se decide a implantar el sistema en la empresa u organización es porque se considera que su rendimiento es óptimo y porque ya ha superado algunas pruebas. Sin embargo, no existe un momento específico de aplicación de pruebas, ya que éstas dependen en gran medida del modelo de ciclo de vida que se haya elegido. Así, por ejemplo, si se eligió el ciclo de cascada con subproyectos, las pruebas se realizan cada vez que se termina la codificación y depuración de uno de esos subproyectos, al final se integran todos y se realiza una última prueba con todos integrados.

En cambio, si se utilizó el ciclo de vida de prototipo evolutivo, entonces se realiza una construcción del prototipo que es evaluada (es decir, probada) por el cliente, de allí se hace un refinamiento del prototipo con base en las observaciones del cliente, y se le entrega de nuevo para que otra vez lo pruebe, y así hasta que el cliente queda satisfecho. Como se observa, la aplicación de pruebas al sistema depende del modelo de ciclo de vida que se haya elegido.

En caso de que exista un sistema anterior operando y se haya previsto usar su información, se deben planear las actividades que llevará a cabo el personal usuario y de informática que efectuará la conversión, se deben asignar sus responsabilidades, el usuario debe dar el visto bueno al plan de conversión y debe validar que la información se traslade en forma íntegra y confiable, el personal deberá recibir capacitación para verificar la conversión.

Otra forma de conversión consiste en la captura masiva de los documentos fuente con el fin de crear las bases de datos que se utilizarán, regularmente estas funciones implican gastos adicionales de renta de equipo y contratación temporal de personal, en ese caso debió haberse presupuestado al inicio del proyecto.

Algunas consideraciones que debe tener en cuenta el equipo del proyecto con las actividades de la implantación independiente del modelo utilizado.

1. La implantación de sistemas debe realizarse toda vez que se haya efectuado satisfactoriamente la prueba integral correspondiente.
2. El equipo del proyecto debe prever las medidas necesarias para el periodo de transición del sistema que se implantará.
3. Se debe preparar un plan de implantación donde se involucre a todo el equipo del proyecto y al usuario.
4. Se debe programar la capacitación integral ⁴⁷ del usuario respecto al sistema a implantar.
5. El proyecto concluye una vez que el sistema haya sido implantado y aceptado por el usuario.

La puesta en operación de un nuevo sistema puede darse a través de diferentes métodos los mas comúnmente utilizados son:

- *Implementación directa.*- Existe un riesgo natural, pueden detectarse errores antes de que surjan problemas serios, con este método no se tiene posibilidad de comparar los resultados del nuevo sistema y el anterior, teniendo que ir ajustando el nuevo sistema según se presente la incidencia, lo anterior podría crear una mala imagen del nuevo sistema y sus desarrolladores, amen de que pone en riesgo su operación.
- *Implementación en paralelo.*- Se procesan ambos sistemas simultáneamente por lo que se tiene la posibilidad de comparar los resultados del sistema anterior con el nuevo, esto garantiza que en caso de problemas significativos se modifique el nuevo sistema y se siga entregando resultados con el sistema anterior, obviamente es mas costoso por la duplicidad de funciones.

Existen otros métodos de implantación (prototipos, modular, fases, etc.), que dependerán de las características del modelo de desarrollo utilizado, además de considerar: medio ambiente de la instalación, compromisos con el usuario, políticas de la empresa, etc.

⁴⁷ Integral se refiere a que cuente con los conocimientos, habilidades y actitudes para con el nuevo sistema.

4.4 Operación y mantenimiento

En esta fase, ya tenemos el proyecto con su calendario, las especificaciones claras, los recursos y personas en situación de trabajo, sin embargo para que un sistema pueda operar se deben tener algunas consideraciones en esta fase tales como: ⁴⁸

1. Se deben diseñar e implantar normatividad para todos los procedimientos de control en la operación de sistemas.
2. Se debe revisar la aplicación de los procedimientos de control y seguridad para el manejo de la información.
3. Se debe prepara un presupuesto de operación para disponer de los recursos de cómputo.
4. Se debe realizar un análisis mensual de los consumos y gastos del recurso informático y determinar sus procedencia respecto al presupuesto aplicado.
5. Se deben diseñar e implementar procedimientos normativos para la autorización de modificaciones temporales y permanentes a los sistemas.
6. Debe integrarse a la documentación de sistemas, todos aquellas modificaciones que se realicen a los sistemas implantados.

Los procedimientos deben estar ubicados de tal forma que el procesamiento pueda realizarse en forma exacta y fluida, las modificaciones al sistema deben ser consideradas sólo si cuentan con la autorización apropiada.

En esta fase destacan tres eventos importantes:

1. Procedimientos de control de operaciones.
2. Control de costos.
3. Modificaciones al sistema.

Procedimientos de control de operaciones. ⁴⁹

Durante las fases de operación y mantenimiento del CVDS los procedimientos y políticas de trabajo deben estar definidos de tal forma que los servicios informáticos pueda realizarse en forma exacta, segura y fluida.

En los procedimientos de control de operaciones se deben incluir controles adecuados para garantizar que los procesos se realicen en la fecha y con las especificaciones definidas, supervisar que se lleven a cabo las medidas de seguridad para la entrada de datos, para la distribución de reportes y el almacenamiento de datos, de manera que sólo el personal autorizado deba intervenir.

⁴⁸ Arechiga, G. Rafael. "Código de Sistemas de Informática" Edit. LIMUSA, 1989

⁴⁹ Aguilar Castillo Gildardo, "Apuntes para la materia Administración de Recursos Informáticos", Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998

Algunas consideraciones:

- Se deben programar las cargas de trabajo en función de la disponibilidad de recursos.
- Definir e implementar controles adecuados para garantizar la continuidad de procesamiento.
- Incluir controles para garantizar la integridad de los datos de entrada.
- Debe incluir controles adecuados de distribución de reportes, de manera que solo la reciba el personal autorizado.
- Debe incluir controles adecuados para garantizar la oportuna entrega de los resultados de procesamiento.
- Control de seguridad para el personal, datos y gente.

Control de costos

El control de costos incluye a los sistemas contables de la organización deben registrar y analizar los costos en que incurre el nuevo sistema una vez que esté operando, se debe evaluar periódicamente si los costos son adecuados, algunas consideraciones:

- Se debe elaborar un presupuesto anual de operación que contenga el recurso económico previsto para el ejercicio de cada una de las partidas presupuestales.
- Los nuevos sistemas de cómputo de la organización deben someterse a un estudio económico de factibilidad.
- Se debe evaluar periódicamente si los costos de operación son los adecuados.
- Se debe incluir un análisis de todas las partidas presupuestales que muestre las variaciones que existen entre los gastos presupuestados y los reales.

Modificaciones a los sistemas en producción.

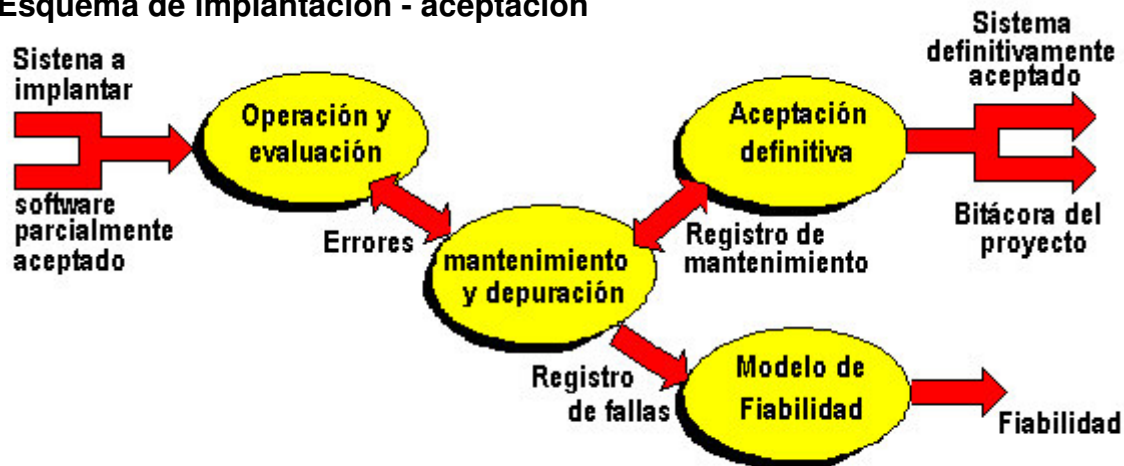
En todos los sistemas de información existe la posibilidad de realizar cambios, ya sea que éstos se originen a solicitud del usuario por presentarse nuevos requerimientos de información, por cambios en los procedimientos y políticas de trabajo, o por que se actualice el hardware y software.

Los sistemas por naturaleza deben ser dinámicos y congruentes con las necesidades de la organización, sin embargo en la mayoría de los centros de cómputo, cuando falla algún programa, o se requiere de un cambio necesario, se hacen reparaciones improvisadas y generalmente sin control.

Por otro lado, es posible que se presenten errores en los programas por haber sido insuficientes las pruebas en su desarrollo o por presentarse alguna condición no prevista que origine la falla. Sin importar lo que motive a efectuar un cambio, será necesario que éste se sujete a un procedimiento estricto de control y seguridad.

Lejos de representar una desventaja, las modificaciones a los sistemas son necesarias, debemos asegurarnos que éstos no caigan en la obsolescencia y cumplan con el objetivo para lo cuál fueron creados.

Esquema de implantación - aceptación



Es muy frecuente recibir la instrucción “corríjase el problema y después veremos que hacer” aunque todos sabemos que una vez resuelto el problema ya nada se hará, por lo que no se documenta la falla, no se registra para su control y no se actualiza la documentación que se ve afectada.

Existen básicamente dos tipos de cambios que se realizan a los programas:

Temporales.- Se identifican porque sólo afectan a la corrida que se está realizando, es decir, una vez que concluye se debe eliminar la modificación, como si nunca hubiese existido, ahí está el riesgo, es importante implementar controles estrictos.

- Si el cambio afecta únicamente parámetros de control, éstos se deberán realizar en una copia y asignarle un nombre que identifique a la entidad como de paso, un cambio de este tipo nunca deberá hacerse en archivos oficiales de producción.
- Si se trata de un programa, de igual modo se deberá realizar en una copia y en una biblioteca alterna a la de producción, se deberá copiar de igual modo, el procedimiento de control para que se direcciones al programa copiado, una vez realizada la corrida se debe respaldar y borrar las copias.

En ambos casos se debe anotar en la bitácora de control la naturaleza de los cambios y los tiempos de proceso.

Permanentes.- Se identifican por que permanecerán en lo sucesivo, por lo que se tendrán que sujetarse a un procedimiento que incluye:

- a) Solicitar al área responsable de los sistemas en producción, la copia de la entidad que se va a modificar a las bibliotecas de desarrollo.
- b) Realizar el cambio solicitado en base a la solicitud del usuario, y aplicar todas las pruebas que se consideren.
- c) Documentar el cambio de acuerdo a procedimiento establecido y afectar los manuales de operación y del usuario que se vieron afectados.
- d) Solicitar al área de operación la implementación de las entidades afectadas en las bibliotecas de producción.

Para las modificaciones del sistema se deben establecer procedimientos para controlarlas y asegurarse de la existencia de un registro cronológico de todos los cambios, se debe evaluar la modificación o cambio propuesto tomando especial consideración si éstas aprobaciones se obtienen del departamento usuario, se sugiere el siguiente formato de control:

				INSTITUTO MEXICANO DE INFORMATICA Subdirección General de Finanzas Jefatura de Servicios Informáticos			
MANTENIMIENTO AL SOFTWARE		Usuario:		Fecha:			
		Reporta: E-mail:		Folio:			
Dirección:			Ciudad:		Teléfono:		
DATOS DEL REPORTE							
Hora:	Aplicación:		Proceso:		<i>Programa:</i>		
Descripción:							
ACCIONES REALIZADAS							
Hora:	Fecha:	Analista que atiende:					
Modificación a:		Descripción:					
Programa fuente							
Parámetros							
Cláusulas de control							
Otros:		Cambio Temporal		Cambio permanente			
NOMBRE Y FIRMA				FECHA	HORA		
Autorizo							
Reviso							

Quando un sistema en operación ha sido modificado, su documentación deberá formar parte integral de ésta e incluir los nuevos requerimientos y las

4. Ciclo de vida de desarrollo de sistemas

modificaciones aprobadas, la actualización a diagramas, manuales y a toda la documentación que se haya visto afectada.

El área de sistemas comprueba que los cambios se realizaron satisfactoriamente, que las pruebas fueron suficientes y que la documentación fue actualizada.

4.5 Control y evaluación

Después de que un proyecto se ha implantado, debe hacerse una revisión de post-implantación para determinar si éste realmente ha cubierto los requerimientos del usuario, medir el grado de satisfacción en términos de objetivos y del análisis costo-beneficio.

Se debe programar una revisión de los resultados de procesamiento por parte del grupo de control de calidad en forma calendarizada, debe contener una evaluación del cumplimiento de los requerimientos y del grado de satisfacción del usuario para determinar si sus necesidades han sido cumplidas en base a la solicitud original.

Esta revisión debe incluir:

- Evaluación de los productos informáticos
- Normatividad respecto a un producto informático
 - Acciones correctivas

Evaluación de los productos informáticos

La evaluación de productos informáticos debe realizarse durante todas las etapas de su desarrollo, para asegurar que el producto terminado sea de calidad, sin embargo, la etapa de evaluación más ardua es la que corresponde a la última fase, es decir, aquella que se realiza una vez terminado el producto.

Para asegurar que dicha evaluación sea satisfactoria, quien desarrolla el producto informático debe tener una serie de controles, conocidos como controles de tecnología informática,⁵⁰ que son los procedimientos que aseguran que los programas operan apropiadamente y que se impidan cambios no autorizados.

Los controles de tecnología informática se incluyen en el diseño, implementación y operación del sistema y por lo general consisten en una combinación de procedimientos manuales y programados, estos controles son importantes porque evitan interrupciones en las operaciones de la organización a causa del producto informático, entre los errores que evita se encuentran.

⁵⁰ Jones, Caper. "Activity-based software costing," Computer, May 1996

- Múltiples reprocesos de las aplicaciones para corregir errores, causando retrasos.
- Afectar la información usada por la administración para operar la organización.
- Permitir la ocurrencia de fraudes, tal como la alteración no autorizada de los registros electrónicos.
- Permitir la divulgación de información confidencial.

Los siete controles primarios de tecnología informática son:

- 1. Controles de implementación.** Diseñados para asegurar que los procedimientos programados sean apropiados y correctamente implantados en los programas computacionales. Controles de este tipo son: *administración del desarrollo del sistema; diseño de sistema y preparación de programas; prueba de programas y sistemas; catalogación.*
- 2. Controles sobre el mantenimiento.** Sirven para asegurar que los cambios a los procedimientos sean diseñados, probados, aprobados e implementados apropiadamente. Por ejemplo: *administración de los cambios a sistemas; prueba de programas.*
- 3. Controles sobre operación del computador.** Permiten asegurar que los procedimientos programados sean aplicados consistentemente durante todo el procesamiento de la información y que se utilicen las versiones correctas de los archivos. Algunos son: *planificación de la carga; preparación y ejecución de trabajos; uso correcto de los archivos de datos; monitoreo de actividades; procedimientos de respaldo y recuperación.*
- 4. Controles sobre la seguridad de los programas.** Aseguran que cambios no autorizados no puedan ser efectuados a procedimientos programados. Estos pueden ser: *software de control de acceso; controles sobre el acceso físico; segregación de funciones.*
- 5. Controles sobre seguridad de los archivos.** Evitan que los cambios no autorizados puedan ser efectuados a archivos de datos. Los cuales son: *software de controles de acceso; acceso a datos en línea; acceso por los programas en producción; archivos fuera de línea; seguridad física.*
- 6. Controles sobre el software de sistemas.** Aseguran que el software de sistemas sea implantado correctamente y protegido contra cambios no autorizados. Estos serían: *informe de trabajos procesados; manejo de archivos; comunicaciones de datos; informe de operaciones; preparación de archivos; seguridad de acceso; registro de bibliotecas.*

7. Controles sobre la conversión de archivos. Aseguran que los datos son convertidos en forma total y exacta desde un sistema antiguo hasta otro nuevo.

Estos controles permiten garantizar la fiabilidad del producto y la seguridad que presenta, de tal forma que la mejor manera de evaluar un producto informático es revisar si tiene todos los controles necesarios y si éstos funcionan en realidad.

Por otra parte, durante la ejecución del producto, o sea su uso o puesta en marcha se puede evaluar adecuadamente, puesto que el usuario es quien mejor puede determinar si un producto informático funciona adecuadamente o no, por lo que resulta conveniente realizar entrevistas periódicas al personal.

De la misma manera se deben realizar pruebas de manera constante para determinar si el producto funciona adecuadamente o si necesita modificaciones y mejoras. Un producto informático siempre es susceptible de mejorar por lo que conforme pasa el tiempo se van realizando versiones más actualizadas y mejor terminadas del mismo.

Normatividad respecto a un producto informático

En cuanto a los estándares de calidad para los productos informáticos se puede decir que existen varias. Por principio de cuentas cuando queramos conocer los estándares para el producto que vamos a realizar, podemos recurrir a las normas ISO que están creadas para determinar aquellos estándares que servirán para determinar si un producto informático es de calidad o no. Estas normas se aplican también a procedimientos y a empresas en general.

Otro punto importante a revisar son las leyes de nuestro país, sobre todo aquellas que hablan sobre los derechos de autor, puesto que todos los productos informáticos se hallan protegidos como propiedad intelectual.

Acciones correctivas

Las acciones correctivas se realizan durante todas las fases del desarrollo de un proyecto. Cuando se realiza el análisis del mismo se determinan las características que tendrá, las cuales cambian al momento de diseñar el proyecto y cuando se está desarrollando el mismo vuelven a cambiar, por lo que todo el tiempo se está trabajando con retroalimentación.

Mientras el producto informático está en proceso de elaboración, estos ajustes y modificaciones son muy válidos, sin embargo, en el momento en que el producto se entrega se supone que está totalmente probado y revisado para su adecuado funcionamiento. Desgraciadamente resulta muy común que después de que el producto se ha implantado se encuentren errores que pueden ser mínimos o muy peligrosos.

En este momento se utilizan diversas acciones correctivas. Por ejemplo, si el problema que se presenta es mínimo, las acciones correctivas se pueden realizar mientras el sistema está funcionando aunque sea parcialmente, pero si los errores son graves debe interrumpirse la utilización del sistema y revisar todo el procedimiento.

Para evitar retrasos en los procesos de la empresa, la acción correcta es correr el sistema en paralelo. Esto significa que si se realiza un sistema informático nuevo para la empresa, se debe poner a funcionar mientras sigue operando el sistema anterior, de esta forma se evitan retrasos en las actividades de la empresa y no se entorpece su buen funcionamiento.

Se supone que si el proyecto se desarrolló adecuadamente durante todas sus fases, será innecesario realizar acciones correctivas al final, puesto que éstas se fueron haciendo a lo largo del desarrollo del proyecto, conforme se presentaban los problemas

Bibliografía de este capítulo:

1. Aguilar Castillo Gildardo, “Apuntes para la materia Administración de Recursos Informáticos”, Facultad de Estadística e Informática, Universidad Veracruzana. México, 1998.
2. Arechiga, G. Rafael. “Código de Sistemas de Informática” Edit. LIMUSA, 1989
3. Bird, Hair, Lamb, *Marketing*, International Thompson Editores, 7a. Ed., 2000
4. Checkland, Peter and Shoges, Jim. *Metodología de los sistemas suaves de acción*, Megabyte – Noriega editores, México.
5. Domingo Ajenjo Alberto, “Dirección y Gestión de Proyectos, un enfoque practico “ Ed. Alfaomega, Madrid España, 2000, pp 294.
6. Gordon B. Davis Margrethe H. Olson, “Sistemas de Información gerencial”, Segunda Ed. McGraw Hill
7. Jones, Caper. “*Activity-based software costing*,” Computer, May 1996
8. Shtub, A., Bard, J. Et Sholomo, G. “*Project Management Engineering, Technology and Implementation*”, Prentice-Hall, 1994.
9. Wilson, Be. *Sistemas: conceptos, metodologías y aplicaciones*, Megabyte-Noriega editores, México, 1990.
10. Zachman, John A. *El modelado de las empresas: la arquitectura de Zachman*.
11. Zachman Institute for Framework Advancement. Estados Unidos, 1999.