

Club de Investigación Tecnológica

Proyectos de Sistemas

**Preparado por: Ignacio Trejos Zelaya
Setiembre, 1989**

**Editado y publicado por Rho-Sigma, S.A.
a nombre del Club de Investigación Tecnológica.
Todos los derechos reservados.
Prohibida la reproducción total o parcial.
San José, Costa Rica
Setiembre, 1989**

Resumen Ejecutivo

La tecnología de la información juega un papel crecientemente estratégico en las organizaciones, las cuales fundamentan cada vez más en los sistemas de información su competitividad y su adaptación a los cambios en el medio. El desarrollo de sistemas de información eficaces requiere de una administración adecuada, que garantice una orientación acorde con los objetivos y estrategias de la empresa, dentro de las limitaciones de recursos y de tiempo.

El desarrollo de un nuevo sistema de información es una actividad no rutinaria en cualquier empresa. Además, un sistema de información introduce modificaciones en la forma de operación de la empresa. Aunque es posible pensar en desarrollar sistemas de información sin seguir metodologías ni planificar, ello aumentaría los costos y los riesgos de fracaso. Por ello es preferible desarrollar los sistemas mediante proyectos.

El presente informe versa sobre la planificación y organización de proyectos de sistemas de información. Su objetivo primordial es contribuir a la comprensión cabal de ese proceso, de manera que las empresas puedan emprender más exitosamente el desarrollo de sistemas de información.

Después de una introducción a los proyectos informáticos, se explican los papeles que desempeñan los diversos componentes de la organización relacionados con tales proyectos. Asimismo, se detalla el importante papel que juega el director del proyecto, con una explicación detallada de sus funciones, responsabilidades y relaciones con el resto de la organización y el equipo del proyecto.

No existe una manera universal de administrar proyectos de sistemas, pero se dan guías para seleccionar herramientas administrativas adecuadas a las características del proyecto. La comprensión del riesgo de un proyecto permite hacer una correcta planificación del desarrollo del sistema. El apéndice contiene un formulario que ayuda a evaluar el riesgo de un sistema, como una herramienta para la evaluación previa y la planificación táctica de proyectos.

Se revisan luego las opciones más comunes para organizar el equipo del proyecto, recomendando pasar paulatinamente de una organización funcional a una matricial. Se ofrecen algunas metas que debe buscar el equipo del proyecto para poder ser eficaz.

Se comentan además los aspectos técnicos que llevan a la construcción del plan del proyecto. Se incluyen consideraciones sobre acciones contingentes así como sobre la documentación y presentación del plan.

Se termina el cuerpo del informe con un capítulo sobre las estrategias no convencionales para desarrollar sistemas. Se dan las características del sistema (regularidad, estructura e impacto), que definen la estrategia más apropiada para desarrollarlo.

Agradecimientos

Se agradece la valiosa colaboración del Ing. Luis Chaves, quien hizo numerosas observaciones de forma y de fondo que contribuyeron a mejorar sensiblemente la calidad de este trabajo. Al Dr. Roberto Sasso y al Lic. Jorge Walter Bolaños agradecemos la lectura concienzuda de los varios borradores que se elaboraron; ellos dieron muchas recomendaciones para mejorar el texto.

Este trabajo es resultado de la experiencia académica y profesional del autor, quien asume responsabilidad por los errores de forma o de fondo que pudiera contener.

Tecnológico de Costa Rica. Ha sido director de Cooperación, profesor y director del Departamento de Computación del mismo Instituto. Anteriormente fue responsable de proyectos de sistemas, analista-diseñador y programador de aplicaciones administrativas, en Guatemala. Actualmente realiza estudios de postgrado en Computación, en la Universidad de Oxford, Inglaterra.

Del autor

Ignacio Trejos Zelaya es ingeniero en Computación Administrativa por el Instituto

Contenido

Página

I ¿Qué es un Proyecto de Sistemas? 1

 1 ¿Qué es un Proyecto? 1

 2 ¿Qué Justifica el Desarrollo de un Sistema? 1

 3 El Ciclo de Vida de los Proyectos 2

 4 El Ciclo de Vida en la Realidad 4

 5 Ventajas y Desventajas de los Proyectos 4

 6 Factores Críticos para el Exito de los Proyectos 5

II El Entorno Organizacional de un Proyecto 6

 1 ¿Quién Dirige el Proyecto? 6

 2 Papel del Usuario 7

 3 Comités Conductores 9

 4 Papel de la Administración Superior 9

 5 Papel del Departamento de Informática 11

 6 Enfoque Recomendado 11

III Un Vistazo a la Administración de Proyectos 13

 1 Factores de Riesgo en un Proyecto 13

 2 Administración del Proyecto Según su Riesgo 14

 3 Importancia de la Planificación y la Organización 14

 4 Problemas Comunes en la Planificación 15

 5 ¿Cuándo debe Planificarse el Proyecto? 16

 6 ¿Quiénes Planifican el Proyecto?..... 18

 7 Notas Bibliográficas 18

IV Organización del Equipo del Proyecto 20

 1 Opciones de Organización 21

 1.1 Organización Funcional 21

 1.2 Organización por Proyecto..... 22

 1.3 Organización Matricial 24

 2 Metas de la Organización 25

 3 Evolución de la Organización Durante el Proyecto 26

V Aspectos Técnicos de la Planificación 30

 1 Estructura del Desglose del Trabajo y Asignación de Tareas 30

 2 Calendarización (Itinerarización) 32

 3 Estimación de Recursos 35

 3.1 ¿Quién hace la Estimación? 36

3.2 Algunas Técnicas de Estimación	36
3.3 Problemas y sus Posibles Soluciones	37
4 Presupuesto	37
5 Otras Actividades que Requieren Planificación	38
6 Contingencia	38
7 Presentación del Plan	40
VI Estrategias para el Desarrollo de Sistemas	42
1 Selección de una Estrategia para Desarrollar el Sistema	42
2 Adquisición de Sistemas Pre-Hechos (Paquetes)	43
3 Sistemas Contratados	45
4 Prototipos y Generadores de Aplicaciones	46
5 Desarrollo Incremental y por Fases	49
6 Desarrollo por el Usuario Final	49
VII Apendice	52
VIII Bibliografía Consultada	54

I ¿Qué es un Proyecto de Sistemas? .c1.

El desarrollo de sistemas de información (o la adquisición de sistemas informáticos) es una actividad innovadora en cualquier empresa ¹. Como se explicó en otro informe del CIT ([Bondu89]), la necesidad de un cambio debe relacionarse con las metas de la empresa. Esto asegura que todos los cambios planificados conduzcan a metas comunes. Los sistemas de información no son la excepción. La formulación del plan maestro de sistemas de información debe formar parte de la planificación general de la empresa.

1 ¿Qué es un Proyecto?.c2.

Un proyecto es una empresa única, no-repetitiva y temporal que se acomete para alcanzar objetivos convenidos previamente, dentro un plazo y presupuesto dados.

Como cada proyecto es único, siempre tiene un elemento de riesgo. La administración de proyectos es el proceso que busca garantizar la consecución de los objetivos del proyecto a tiempo y dentro del presupuesto, a pesar de los riesgos y problemas que puedan ocurrir en el camino.

Aunque cada proyecto es único, se han desarrollado algunas técnicas administrativas que son aplicables a los proyectos en general, y para los de desarrollo de sistemas de información, en particular.

La administración de proyectos comprende ([Rosenau84], [Page-Jones85]):

- **Definir** los objetivos del proyecto.

¹ Se usa el término "empresa" para denotar cualquier organización privada o pública (compañía, institución, sociedad sin fines de lucro, etc.).

- **Planificar** las tareas intermedias que conducen a la satisfacción de los objetivos del proyecto.
- **Organizar** la asignación de personas a las tareas identificadas por el plan del proyecto, así como definir líneas de mando y funciones.
- **Integrar** al equipo del proyecto eficazmente en las actividades técnicas y administrativas del proyecto.
- **Controlar** ("monitorear") el avance del proyecto, en particular, si se desvía del plan. También incluye la recopilación de estadísticas sobre el trabajo y el tiempo dedicados a lo largo del proyecto.
- **Corregir**, que consiste en hacer los cambios que atiendan las desviaciones respecto del plan.
- **Completar**, es decir, asegurar que el trabajo realizado está conforme con la última definición de lo que debía hacerse, así como atar cualquier cabo que haya quedado suelto.

2 ¿Qué Justifica el Desarrollo de un Sistema? .c2.

Para que un proyecto sea un éxito, el sistema resultante debe ser de alta calidad técnica, así como brindar una mejora eficaz a las actividades que realiza el usuario. Para que sea eficaz y factible, el proyecto debe ofrecer beneficios que excedan sus costos y no debe conllevar costos mayores que los recursos disponibles.

Un proyecto brinda beneficios si resuelve un problema empresarial o si permite aprovechar una oportunidad. Los beneficios son mejoras en la competitividad de una empresa y se pueden clasificar en:

- incremento de los ingresos,
- reducción de los costos,
- mejoría en los servicios al cliente,

- lograr una diferenciación de mercado para un producto o servicio
- dar valor agregado a un producto o servicio, y
- cumplir con un requisito comercial.

3 El Ciclo de Vida de los Proyectos .c2.

Se han elaborado modelos para ordenar el proceso de desarrollo de sistemas. Uno de los más comunes es el del "ciclo de vida", también llamado "de la cascada". El "ciclo de vida", que se expone seguidamente, fue elaborado como parte de la iniciativa STARTS del Reino Unido ([NCC87]).

Previo al desarrollo del sistema propiamente, se hace una **evaluación inicial**, en la cual se identifican las necesidades de la empresa, así como los procedimientos, alcance, restricciones y problemas por resolver. Incluye un enunciado de los objetivos del sistema, las opciones de desarrollo, así como una estimación preliminar de sus posibles beneficios, costos y riesgos.

Esta evaluación inicial debe desprenderse del plan maestro (ver [Bondu89]), detallando y actualizando las consideraciones, incluidas en ese plan, acerca del posible proyecto.

En esta evaluación inicial también conviene:

- valorar las condiciones del medio en lo que respecta a la tecnología, competencia, restricciones legales, etc.
- valorar las condiciones de la empresa: recursos disponibles, capacidad de cambio, personal, experiencia, etc.

La administración superior decide si es procedente planificar un proyecto, para así evaluar su factibilidad, si debe buscarse una alternativa manual, o simplemente cancelar cualquier esfuerzo adicional.

Con miras a lograr los objetivos del proyecto, la fase de **planificación** comprende:

- la identificación y evaluación de opciones,
- la determinación provisional de costos, beneficios y riesgos,
- la asignación de recursos, y

- la itinerarización de actividades ².

Además, se acostumbra indicar:

- las herramientas, métodos y estándares por usar,
- las estrategias para garantizar la calidad, realizar la verificación y administrar las configuraciones del producto,
- los criterios de aceptación del sistema por la administración superior y el usuario.

Un grave error que se comete comúnmente luego de esta fase es obviar las fases de especificación y diseño, para producir algo tangible (programas) como respuesta a presiones superiores, pues se parte del erróneo supuesto de que ya existe un diseño factible del producto.

El desarrollo, propiamente dicho, se compone de las siguientes fases ([NCC87]):

- **especificación de requerimientos** (análisis),
- **diseño arquitectónico**,
- **diseño detallado**,
- **codificación y prueba de unidades** (programación),
- **integración del sistema**,
- **aceptación e instalación**,
- **evolución** (mantenimiento), y
- **terminación**.

Es conveniente definir las fases en términos de sus **productos**. Estos, al ser evaluados satisfactoriamente, indican que se completa una fase, se convierten en **distribuibles** y forman la **base** para el trabajo de la siguiente fase.

Los productos marcan, pues, los **hitos** que indican el progreso, eficazmente evaluable, del proyecto. Esta es la única manera de evitar la trampa de "el proyecto ya está completo en un 95%".

²Por itinerarización se entiende el ordenamiento de las actividades en el tiempo

Una actividad en el desarrollo de sistemas está **terminada, por iniciar o incompleta**, no hay estados intermedios. El proyecto, a su vez, estará terminado o incompleto, **nunca** estará "terminado en un 95%". Muchísimos proyectos informáticos han dedicado no menos de un 50%

del esfuerzo total a pasar de un "95%" a la terminación real del sistema.

La figura 1 muestra la relación "insumo-producto" entre las fases de desarrollo de un sistema.

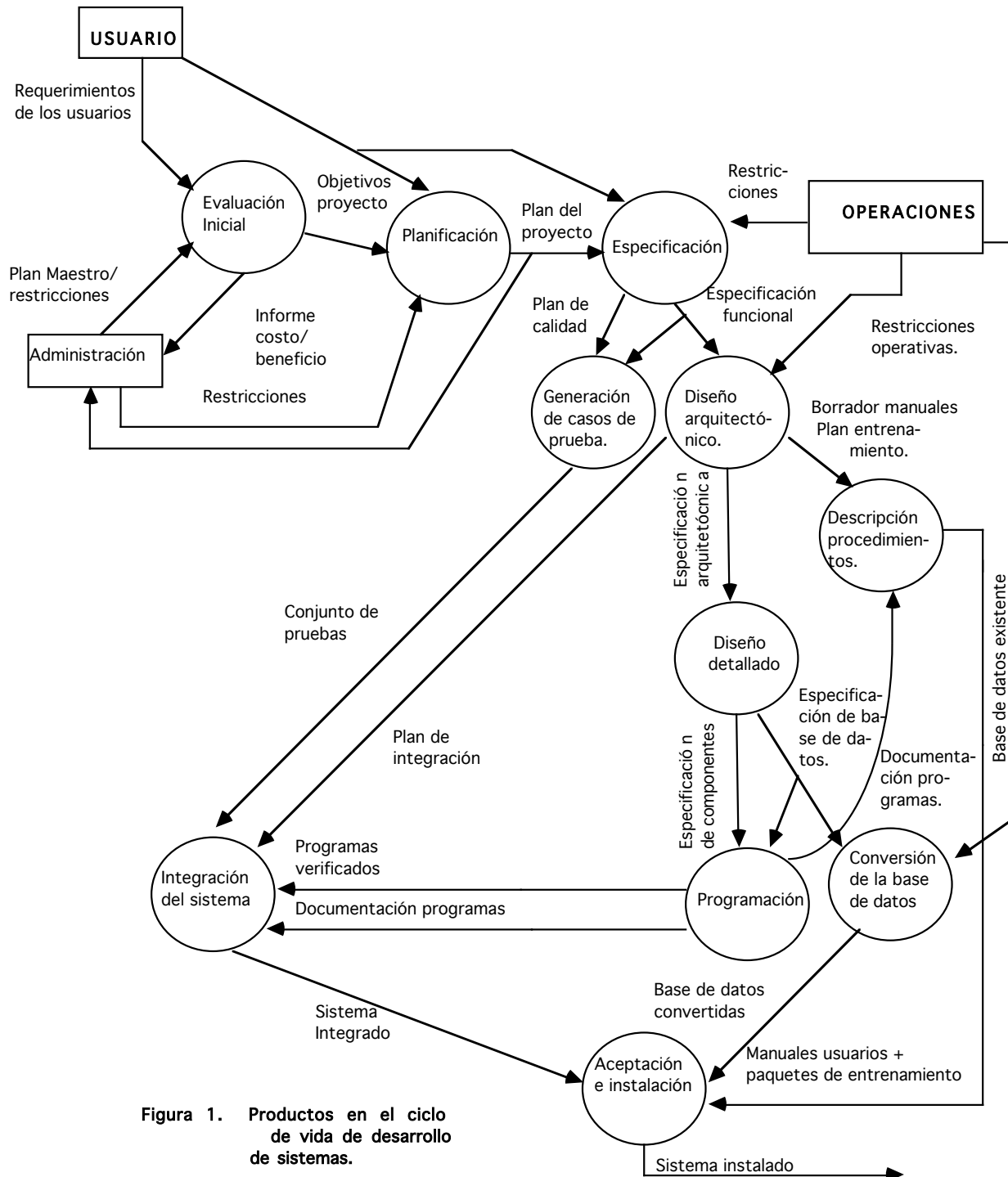


Figura 1. Productos en el ciclo de vida de desarrollo de sistemas.

4 El Ciclo de Vida en la Realidad .c2.

Es común que ocurra alguna de las siguientes situaciones cuando se conduce un proyecto:

- necesidad de hacer algún trabajo exploratorio en aspectos que corresponden a la siguiente fase, antes de terminar "formalmente" una fase (por ejemplo: para determinar costos o factibilidad técnica);
- se manifieste un problema ocasionado por lo que se ha hecho en una fase previa;
- las necesidades y los objetivos del usuario pueden variar durante un proyecto, provocando cambios en la especificación de requerimientos (y en fases subsiguientes);
- puede usarse estrategias de desarrollo incremental o por prototipos (ver capítulo 6).

El modelo de desarrollo presentado en la sección 1.3 no es único, ni debe imponerse "por decreto" ([Yourdon82]). El modelo es una guía general, que debe adaptarse a cada ambiente. Los directores de proyecto y los técnicos pueden resistirse si sienten que no tienen suficiente participación en el asunto. Hay que evitar la táctica de entrenar superficialmente, en una metodología o técnica, al equipo del proyecto, pues es muy probable que al poco tiempo tendrán tropiezos.

Al introducir cambios e innovaciones en el enfoque utilizado para desarrollar sistemas, puede ensayarse con "proyectos piloto". Los participantes en tales proyectos brindarían consejo y orientación realistas a los siguientes grupos que adopten el nuevo enfoque. Para asegurar el éxito y aumentar la confianza de la organización, conviene que los proyectos piloto sean de pequeño tamaño y complejidad.

[De Marco82] sugiere no tener una metodología general para los proyectos, sino más bien una metodología específicamente ajustada para cada proyecto. El punto de partida puede ser el modelo de desarrollo prevaleciente (como el ciclo de vida expuesto antes). La metodología ajustada será el modelo de desarrollo que se seguirá para desarrollar el sistema. Este modelo será uno de los primeros resultados de la fase de planificación. De Marco también sugiere contar con estándares como: plantillas ³ de actividades y de distribuibles ⁴, muestras de distribuibles y "esqueletos" de modelos de desarrollo.

Finalmente, no hay por qué insistir en una cadena estrictamente secuencial de actividades para el proyecto, pues el entorno en que se desarrollará es dinámico (salvo contados casos). De hecho, muchos sistemas se componen de subsistemas y es posible, por ejemplo, iniciar el diseño y construcción de un subsistema ya especificado antes de terminar el análisis (especificación) de todos los subsistemas que componen el sistema total.

5 Ventajas y Desventajas de los Proyectos .c2.

A pesar de conllevar un costo adicional en su administración, normalmente resulta preferible desarrollar un sistema de información como un proyecto. El proyecto permite tener una administración centralizada, relaciones formales con la organización, un mayor control del costo y de los recursos asignados, así como economías de tiempo.

³ "machotes" o formatos predefinidos que se utilizan como punto de partida para la elaboración de un documento, programa, diseño, etc.

⁴ Los "distribuibles" son los productos (documentos, programas) que van resultando de cada fase del desarrollo. Una vez revisados se pueden entregar a los diversos participantes del equipo del proyecto.

En efecto, si las actividades de desarrollo se llevan a cabo mediante un proyecto, es posible crear:

- la reunión de habilidades provenientes de diversas unidades de la empresa;
- canales de comunicación y formas de trabajo que se ajustan a la tarea que debe realizarse;
- una división de trabajo y responsabilidad acorde con el tamaño y complejidad del proyecto;
- un proceso de decisión simplificado y ágil;
- un medio para satisfacer las demandas de participación de los empleados en la toma de decisiones;
- énfasis en los objetivos.

Una alternativa podría ser administrar el desarrollo mediante comités de participantes (por ejemplo: usuarios, sistemas de información, subcontratistas). Esto conduce a una inevitable disipación de esfuerzos y responsabilidades, debido a que el trabajo representa sólo una fracción de las responsabilidades de cada miembro del comité. Además las decisiones y soluciones negociadas son más difíciles de alcanzar (lo que incrementa el costo). El control del presupuesto también se dificulta.

Además del costo adicional de administrarlo, el proyecto tiene costos de establecimiento y de terminación, así como los problemas asociados de reclutar y reintegrar personal a su puesto original. De hecho, en nuestro medio existe la tendencia a prolongar más allá de lo necesario la presencia del personal en el proyecto.

Por otro lado, los proyectos cortan a través de varios departamentos, lo que puede causar recelo y antagonismo en el resto de la organización. El personal del proyecto puede llegar a tener lealtades divididas entre su departamento y el proyecto. Con todo, las ventajas superan normalmente las desventajas, salvo en el caso de sistemas absolutamente triviales.

6 Factores Críticos para el Éxito de los Proyectos .c2.

En [Slevin87] se presentan los **factores críticos para el éxito** de un proyecto. La propuesta de Slevin y Pinto se basa en un estudio que realizaron sobre 418 proyectos de diversa índole. Los factores se presentan a continuación:

- **Misión del proyecto.** Claridad inicial de los objetivos y la dirección del proyecto.
- **Respaldo de la administración superior.** Disposición de la alta administración en cuanto a proveer los recursos, la autoridad y el poder necesarios para el éxito del proyecto.
- **Plan del proyecto.** La especificación detallada de las acciones requeridas para la implementación del proyecto.
- **Consulta con el usuario** (o el cliente). Comunicación activa con todas las partes afectadas por el proyecto.
- **Recursos humanos.** Reclutamiento, selección y entrenamiento del personal necesario para el equipo del proyecto.
- **Tareas técnicas.** Disponibilidad de la experiencia y la tecnología necesarias para la realización de acciones específicas.
- **Aceptación por el usuario.** La "compra" de los resultados del proyecto por sus usuarios.
- **Control y retroalimentación.** Provisión oportuna de información de control en cada etapa del proceso de implementación.
- **Comunicación.** Proveer los datos necesarios para los actores claves del proyecto, así como una red de comunicación que facilite la implementación.
- **Manejo de problemas.** Capacidad para atender crisis inesperadas y las desviaciones del plan.

De estos factores de éxito críticos, los tres primeros (misión, apoyo de la administración superior, plan) son de carácter **estratégico**, pues tienen que ver con los objetivos y cómo alcanzarlos. Los restantes factores son **tácticos** y enfatizan el uso que se dará a los recursos humanos, técnicos y financieros para alcanzar los fines estratégicos ⁵.

La importancia de la estrategia es mayor en las etapas iniciales de un proyecto (evaluación inicial, planificación, especificación y diseño arquitectónico) y va disminuyendo progresivamente en las siguientes etapas. La táctica crece en importancia a partir de la etapa de especificación, alcanzando su máximo durante la fase de integración.

⁵ Los términos "táctico" y "estratégico" se usan aquí en el contexto de cada proyecto, no con la amplitud con que se los usa en [Bondu89], donde el contexto está formado por los sistemas de información de toda la empresa.

II El Entorno Organizacional de un Proyecto

.c1.

La inserción del proyecto dentro de la planificación estratégica de la empresa, el apoyo decidido de la administración superior y la participación del usuario son algunos de los factores críticos para el éxito de un proyecto informático.

La función informática, al igual que la de servicios administrativos, sirve a **toda** la empresa. Por esta razón, no es recomendable colocar el departamento de informática bajo otra área funcional, ya que se corre el riesgo de centrar sus servicios en esa área (ejemplo de ello era la costumbre de ubicar el "centro de cómputo" bajo la jefatura financiero-contable).

Se ha comprobado que situar al departamento de informática cerca de los niveles de gerencia superior mejora sustancialmente la administración de los asuntos relacionados con sistemas de información.

Un proyecto es una estructura temporal dentro una organización. Esto necesariamente creará canales de comunicación y de autorización inusuales dentro de la organización.

En este capítulo se examinan más de cerca algunas opciones de organización de la empresa en relación con el proyecto, que permiten mantener un control adecuado sobre el proyecto, pero previenen los excesos administrativos que impedirían su avance.

Es muy común que en nuestro medio se den los siguientes problemas:

- los usuarios no conocen o no comprenden plenamente los beneficios que los sistemas de información pueden aportar para la mejora de sus funciones;

- nuestros técnicos, o los usuarios, rehúyen a asumir la responsabilidad por el fracaso de los proyectos;
- la mayoría de los sistemas de información resultan caros, se entregan tarde y no son aceptados por los usuarios;
- se asignan responsabilidades técnicas o administrativas a personal con preparación deficiente o experiencia insuficiente;
- los objetivos de los proyectos no se derivan de los objetivos el plan estratégico de la empresa.

Una modificación de la situación prevaleciente hacia una en la cual sea el usuario quien adquiera la principal responsabilidad por el éxito del proyecto, requerirá cambios profundos en la cultura de la empresa y en su estructura administrativa. Es obvio que requerirá también cambios en la actitud del departamento de informática, así como en la forma de organizar su trabajo.

En contraposición con la común "cultura del incumplimiento" que se da en nuestro medio, la alta gerencia debe estimular que en la empresa surja una "cultura por el éxito", que se logrará gracias a la colaboración que brinden los usuarios, la alta gerencia y los técnicos en informática para alcanzar el éxito de cada proyecto.

A continuación se describe un modelo organizacional que es válido para empresas de gran tamaño. Sin embargo, las empresas medianas y pequeñas pueden adaptar el modelo de acuerdo con sus recursos disponibles, al reunir en una sola persona o entidad, según corresponda, varias de las funciones y responsabilidades que se indican.

1 ¿Quién Dirige el Proyecto? .c2.

El director del proyecto es la persona responsable por el éxito (o fracaso) del proyecto. El director tiene responsabilidad y autoridad respecto al alcance de los objetivos dentro del tiempo y de los recursos asignados. Se requiere que conozca la cultura de la empresa, la problemática a que se dirige el sistema y las generalidades de la tecnología informática por utilizar.

El director es el factor individual más importante para el éxito de **cualquier** proyecto. El director debe tener habilidades administrativas que le permitan manejar eficazmente al equipo del proyecto, así como los recursos materiales y el tiempo asignados.

En detalle, sus responsabilidades administrativas incluyen:

- Desarrollo, comunicación y mantenimiento del plan general del proyecto. Esto implica vincularse con los directores de los departamentos involucrados (incluidos otros departamentos además del usuario principal y el de informática ⁶.
- Controlar el avance del proyecto contra el plan.
- Dirigir la evaluación y administración del riesgo.
- Proponer cambios en el proyecto (recursos, límites, objetivos, etc.).
- Tomar decisiones correctivas.
- Dar cuenta de todos los costos y beneficios.
- Administrar los procedimientos generales para el control de cambios.
- Informar oportuna y precisamente al "comité conductor" sobre el avance del proyecto. Así, ante conflictos o problemas, el comité podrá ejercer su potestad.

⁶ Aquí se llama "departamento de informática" o de "sistemas de información" a la unidad encargada de la tecnología informática en la empresa.

Las principales responsabilidades del director son:

- la satisfacción de los objetivos de la empresa
- la correcta y completa satisfacción de usuario.

En cierta medida el director se convierte en un **agente** del usuario, que vela por sus intereses. Para que sea exitoso, el director debe contar con el respaldo decidido de la administración superior y de los departamento usuarios involucrados en el proyecto.

El director es también responsable por verificar la calidad de la contribución técnica del departamento de informática al proyecto. La responsabilidad técnica del director es asegurar que el producto entregado conforma las especificaciones y el nivel de servicio acordados.

Los siguientes factores deben tomarse en cuenta para cumplir con su responsabilidad técnica:

- Asegurar que los objetivos y el plan del proyecto se conocen en el departamento de informática, y que la contribución de éste es comunicada, entendida y convenida.
- Planificar y coordinar la contribución del departamento de informática.
- Controlar el avance de la contribución del departamento de informática.
- Asegurar que las contribuciones técnicas están conforme a los estándares y prácticas vigentes.
- Verificar la calidad del trabajo provisto por técnicos externos o compañías contratadas ⁷.

⁷ Sea porque complementan las labores del departamento de informática o porque ejecutan todo el trabajo técnico.

Uno de los mayores problemas que se enfrentan en nuestro medio es la **carestía de directores de proyecto** con las características definidas anteriormente. Es muy frecuente que se asigne la dirección del proyecto a una persona sin experiencia administrativa o técnica. Las empresas deben propiciar la formación de futuros directores de proyecto mediante la capacitación de personal y la asignación de los prospectos ⁸ como miembros de equipos de proyecto, en funciones de asistencia al director. Antes de promover un director de proyecto a un puesto de mayor jerarquía, la empresa debe haber preparado un sucesor idóneo.

Si la empresa está por establecer una mayor disciplina en la administración de proyectos y no cuenta con directores entre su personal, es conveniente pensar en reclutar directores de proyecto experimentados que provengan de otras empresas.

2 Papel del Usuario .c2.

Varios estudios han encontrado que el usuario juega papeles determinantes para el éxito de un sistema de información ([Boland87], [Wilkinson87]).

Al usuario le agrada involucrarse en el desarrollo de sistemas. Sin embargo, la costumbre ha sido que los desarrolladores de sistemas vean al usuario como un agente extraño o un "mal necesario". Ésta es una situación que debe corregirse, pues en nada beneficia al desarrollo de buenos sistemas de información.

Los usuarios deben llegar a sentir que tienen influencia real sobre los sistemas. Más que eso, **el usuario ha de ser el propietario total del sistema**. El usuario hace la inversión, participa decisivamente en el proyecto y adquiere responsabilidad por su éxito.

Es ideal que los usuarios formen parte del equipo del proyecto, pues ellos facilitan información inmediata acerca de sus necesidades. Esto permite que el personal de informática comprenda mejor los problemas que el sistema debe resolver. Los usuarios miembros allanan la instalación y operación del sistema. Además, las relaciones departamento usuario-departamento de informática mejoran, facilitando el trabajo futuro conjunto.

Hay varios niveles posibles de participación del usuario:

- **consultiva**: se le consulta sobre asuntos relacionados con el sistema, pero no toma ninguna decisión;
- **participativa**: tiene una relación de trabajo cercana con el equipo del proyecto, posiblemente como miembro (no líder) del mismo;
- **directiva**: tiene mucha influencia, toma decisiones y es responsable por su implementación.

Hay varios papeles que el usuario puede desempeñar, con el fin de participar efectivamente en la concepción, conducción, realización y operación de un proyecto informático. Esos papeles se describen a continuación.

⁸ Posibles futuros directores de proyecto.

El **patrocinador** es un administrador superior de la empresa, quien es experimentado e influyente. Él es quien usualmente propondrá el proyecto al comité ejecutivo de automatización, y tiene un genuino interés porque el sistema de información beneficie a las áreas funcionales usuarias (algunas de ellas se encuentran, posiblemente, bajo su línea de mando). Sus áreas de responsabilidad incluyen:

- reforzar, en todas las partes, la comprensión de su co-responsabilidad en el éxito del proyecto;
- asegurar la claridad y el acuerdo en cuanto a objetivos y responsabilidades;
- establecer un comité conductor para el proyecto;
- motivar a los administradores de los departamentos usuarios a proveer los recursos del proyecto que les atañen;
- responsabilidad ejecutiva por el éxito del proyecto;
- influir en el nombramiento de un director de proyecto idóneo, con la personalidad, conocimiento, experiencia y habilidades para tener éxito, como agente del patrocinador, en la conducción cotidiana del proyecto.

Idealmente, el **usuario especificador** es aquel que tiene un conocimiento profesional profundo del área funcional involucrada y experiencia práctica significativa en ella. Habrá tantos de ellos como haya unidades involucradas en el sistema. Es deseable que el usuario especificador sea un director de unidad o alguno de sus jefes intermedios.

En nuestro medio es muy común asignar personal poco experimentado en la función de usuario especificador. La excusa para esto es que "ellos tienen más tiempo disponible". El resultado inevitable será un sistema de información con una especificación inadecuada, lo que aumenta la probabilidad de fracaso del proyecto.

El **usuario final** es aquel personal que operará físicamente el sistema y que trata directamente con las entradas y salidas del mismo. Este usuario es el sujeto de las consideraciones ergonómicas y cognitivas del sistema.

El **generador de entradas** es aquel trabajador que produce los documentos que sirve de entrada al sistema, aunque quizás nunca interactúe directamente con el mismo.

El **receptor de salidas** es quien recibe los documentos producidos por el sistema y, con base en ellos, realiza acciones o toma decisiones. Es posible que jamás llegue a interactuar directamente con el sistema.

3 Comités Conductores .c2.

El papel del **comité conductor** es crear un ambiente favorable para el éxito en el logro de los objetivos del proyecto, a tiempo y dentro del presupuesto. Además sirve como enlace entre los usuarios y el comité ejecutivo de automatización.

Todo proyecto importante debería tener un comité conductor, aunque podría establecerse un solo comité conductor para varios proyectos dirigidos a una misma área funcional de la empresa.

El comité conductor del proyecto consiste de un representante de cada área usuaria importante del proyecto, así como del director de informática. El patrocinador convoca y preside al comité. El director del proyecto forma parte del comité conductor (con voz, pero sin voto) y le rinde informes.

Los representantes del usuario bien pueden ser los usuarios especificadores. A ellos hay que darles la potestad de tomar decisiones en las reuniones del comité, con el acuerdo de que serán aceptadas por los jefes de las áreas funcionales. De otra manera los retrasos serán excesivos.

El comité conductor se reúne con regularidad, pero más frecuentemente durante la especificación del proyecto y las fases de validación y aceptación del sistema construido.

El director del proyecto tendrá la responsabilidad de presentar, en cada reunión, un enunciado de los logros a la fecha, el trabajo en curso, estados que comparen la ejecución contra el calendario y los gastos realizados contra el presupuesto. El director del proyecto también tiene la responsabilidad de alertar al comité conductor sobre cualquier problema o conflicto de intereses que surja, a fin de que el comité pueda ejercer sus atribuciones.

Algunas de las siguientes acciones correctivas pueden emanar del comité conductor:

- resolver conflictos entre áreas usuarias;
- iniciar cambios en el equipo del proyecto;
- replanificar las tareas por realizar;
- cambiar el calendario o el presupuesto;
- revisar los objetivos por lograr;
- ordenar que se rediseñe el sistema;
- detener el proyecto.

El comité conductor ha probado ser una herramienta eficaz para reducir el riesgo de los proyectos informáticos. En algunas organizaciones se **omite** el empleo de tal comité para los proyectos de menor envergadura que afectan únicamente a una área funcional; en estos casos, el patrocinador resuelve los conflictos y recibe los informes del director del proyecto.

El contar con un comité conductor tiene las siguientes **ventajas**:

- los ejecutivos conocen mejor los factores que afectan el logro de los objetivos,
- brinda un espacio para la resolución de conflictos,
- permite la interacción entre los ejecutivos y los responsables de proyecto, y
- fomenta una toma de decisiones participativa.

Por otro lado, se tienen las siguientes **desventajas**:

- el comité puede aislar al director de informática,
- depende mucho del apoyo e interés que manifieste la administración superior,
- los participantes evitan comprometerse en decisiones difíciles, y
- puede darse una degradación en la toma de decisiones.

4 Papel de la Administración Superior .c2.

La participación de la administración superior en establecer y hacer cumplir una política informática es un factor crítico para el éxito de los sistemas de información de la empresa.

El **comité ejecutivo de automatización** guía y coordina el desarrollo de sistemas, para asegurar que se dará alta prioridad a aquellos trabajos de alto valor para la empresa (y para amortiguar la presión que ejercen algunos sectores de la empresa por hacer trabajos urgentes de prioridad baja).

Es este comité quien propone las pautas para la formulación del plan estratégico de sistemas de información y del plan maestro. El comité asegura que la estrategia informática y el plan maestro están acordes con la estrategia empresarial.

Este comité está constituido por representantes de la alta dirección de todos los departamentos o divisiones con intereses en los sistemas de información. El comité sería dirigido por el gerente general o por el subgerente responsable por el área informática. El director de informática es el secretario del comité y no tiene voto. Nolan recomienda que sus miembros sean menos de diez ([Nolan82]).

La responsabilidad del comité ejecutivo de automatización es revisar y controlar el alcance de la organización informática y de sus actividades. Para ello, debe:

- establecerle metas de largo plazo (3 a 5 años);
- revisar y aprobar el plan maestro informático;
- fijar presupuestos generales de los servicios informáticos para sus usuarios,
- dar una guía global a la función informática,
- actualizar el plan maestro de sistemas de información.

El departamento de informática les proveerá información preliminar sobre estimaciones de costo y tiempo, así como sobre las capacidades y limitaciones de la tecnología informática. Cuando hay problemas de falta de credibilidad o falta de experiencia en el departamento de informática, el comité ejecutivo de automatización puede recurrir al criterio de consultores en informática que cuenten con probada confiabilidad.

Este comité es el que establece las políticas operativas para los sistemas de información, por ejemplo:

- centralizar vrs. descentralizar;
- si las microcomputadoras serán controladas por el usuario o por el departamento de informática;
- si la empresa usará paquetes existentes, contratará servicios externos o desarrollará sus propios sistemas;
- determinar los niveles de aprobación que requerirán proyectos de distintos tamaños,

- en general, definir los niveles de autoridad y responsabilidad.

El comité debe examinar la eficacia del departamento de informática en términos del avance total de sus proyectos y la calidad de los servicios que presta. Así evaluará si la empresa está aprovechando o recuperando su inversión en relación con sus planes, sus competidores y el estado actual de la tecnología.

El comité ejecutivo de automatización tiene varias funciones relativas a la autorización y control de los proyectos individuales, entre ellas:

- autorizar los proyectos cuyo costo estimado exceda cierto límite;
- establecer y revisar las prioridades de todos los proyectos aprobados;
- aprobar los presupuestos y calendarios de los proyectos prioritarios, así como cualquier cambio a éstos;
- revisar - a alto nivel - el avance de los proyectos más significativos, para ello contará con el auxilio del comité conductor;
- cancelar proyectos que resulten no rentables;
- nombrar patrocinadores para proyectos aprobados.

Algunas empresas han creado el puesto de "**gerente de información**"⁹, para asegurar la planificación y control integrados de la tecnología informática y los sistemas de información. El gerente de información tiene responsabilidad ejecutiva por la planificación estratégica y táctica de los sistemas de información. Él da también un control gerencial a los proyectos en desarrollo, pero **no** se entiende con los detalles técnicos de los proyectos, ni con el mantenimiento u operación de los sistemas existentes.

⁹ Llamado "director de sistemas de información" en [Bondu89]. Su nombre en inglés es "Chief Information Officer".

En caso de existir, el gerente de información forma parte del comité ejecutivo de automatización, así como de los comités conductores. Cuando los comités conductores no existen, el patrocinador y el gerente de información asumen las funciones y responsabilidades descritas para esos comités.

5 Papel del Departamento de Informática .c2.

Además de los cambios de actitud - en cuanto a las relaciones con el usuario - mencionados previamente, el departamento de informática deberá ajustarse, en diversas formas, para acoger la realización de proyectos.

El papel del **director de informática** es dar apoyo y orientación a los directores de proyecto, en lo relativo al componente informático del proyecto. Su apoyo, directo o mediante el **director de desarrollo de sistemas**¹⁰, puede manifestarse de diversas maneras:

- asistir en la resolución de conflictos;
- revisar constructivamente los conceptos, avance, y problemas técnicos;
- provisión de estándares y buenas prácticas de trabajo;
- velar porque se dé control de calidad, separado del proyecto, en momentos apropiados;
- reconocer que la administración de proyectos es un proceso de aprendizaje continuo.

La **unidad de operaciones**¹¹ se verá afectada por el proyecto durante el desarrollo, por lo que deberá coordinarse con ella lo que sigue:

¹⁰ Si es que existe. La unidad de desarrollo de sistemas se encarga de coordinar los aspectos técnicos del desarrollo de sistemas de información.

¹¹ Esta unidad se encarga de operar los sistemas de información que resultan de un proyecto. Típicamente aparece en departamentos de informática que originalmente fueron centralizados.

- adquisición de equipo, suministros y software;
- requerimientos de recurso computacional (tiempo, almacenamiento, periféricos);
- cargas de entrada de datos;
- requerimientos para comunicación de datos
- asignación de prioridades en el uso del equipo.

Se identificarán, con anticipación, el equipo, software y ayudas especiales necesarios para operar el sistema final. Asimismo, habrá de construirse un calendario para su adquisición e instalación. También es importante contemplar el entrenamiento del personal que operará el sistema (que no necesariamente son usuarios finales).

El proyecto deberá coordinarse con **otros proyectos en curso** o con **sistemas existentes**. Esto puede requerir la replanificación de algunos proyectos y la resolución de conflictos provocados por:

- competir por los mismos recursos técnicos;
- prioridades en la disponibilidad del computador;
- formatos o contenidos distintos en los datos;
- calidad de datos recibidos inaceptable;
- calendarios inaceptables.

También pueden presentarse conflictos entre el proyecto y otras actividades, causados por:

- desacuerdo en cuanto a estándares;
- no aceptación de software común;
- programas utilitarios inapropiados;
- mantenimiento del diccionario de datos;
- opiniones antagónicas a las de los especialistas

El director de informática es quien resuelve estos conflictos, en busca de los mejores intereses de la empresa, en lugar del interés de las partes.

6 Enfoque Recomendado .c2.

En nuestro medio es rara la empresa que ha logrado una disciplina para la administración del desarrollo de sistemas. La ausencia de tal disciplina, así como la falta de claridad en cuanto a los papeles que desempeñan diversos componentes de la empresa - en relación con los sistemas de información -, son fuente de no pocos problemas (y fracasos) en los proyectos informáticos.

En este capítulo se han descrito brevemente los papeles que juegan los actores del proceso de desarrollo de sistemas de información. Ahora bien, la descripción de cargos o de entes colectivos puede resultar excesivamente voluminosa u onerosa para la mayoría de las empresas de nuestro medio. Se dan algunas recomendaciones sobre maneras en las cuales un aparato administrativo más reducido puede producir resultados satisfactorios. Se advierte que las funciones y responsabilidades descritas antes mantienen plena validez.

El nivel ejecutivo más alto no puede soslayar las responsabilidades que se describieron para el comité ejecutivo de automatización. Los niveles gerenciales deben involucrarse porque la tecnología informática puede mejorar su proceso de toma de decisiones, aumentar la competitividad de la empresa en el medio, y estimular la flexibilidad y capacidad de adaptación de la empresa a los cambios en las demandas del medio. El mismo tamaño del presupuesto de la función informática puede llegar a ser considerable y justificar la participación de la administración superior en su dirección estratégica.

La definición de políticas y estrategias informáticas es, pues, responsabilidad de la administración superior. En una empresa mediana, esta responsabilidad debe existir al nivel superior, aún cuando el comité ejecutivo de automatización no esté formalmente constituido. Este papel puede ser desempeñado por el gerente general, con la asistencia de sus subalternos inmediatos y con la asesoría del director del departamento de informática.

El éxito de un proyecto depende, en buena medida, del convencimiento que tenga la administración superior sobre la necesidad de emprenderlo, así como del apoyo que le brinde a lo largo de su desarrollo. El patrocinador juega el importante papel de hacer manifiesto ese apoyo, además de influir al resto de la empresa para que brinde colaboración efectiva al proyecto.

Cuando la empresa es mediana, resulta suficiente que el patrocinador asuma la mayoría de las responsabilidades descritas para el comité conductor. En caso de ocurrir conflictos, el patrocinador puede convocar a reuniones *ad-hoc* en las que participan los directores de las áreas usuarias; en caso de no resolverse ahí el conflicto, el problema es llevado a consideración de la administración superior (al comité ejecutivo de automatización, si éste existe).

Por otro lado, es conveniente asignar **un solo proyecto a la vez** como responsabilidad del director. De otra manera, se corre un alto riesgo de que el director del proyecto diluya su atención en muchos asuntos. La empresa debe tener en cuenta que la formación de directores de proyecto es una responsabilidad continua.

Uno de los factores que aumentan el riesgo de un proyecto es el **tamaño** del sistema por desarrollar. La empresa puede evitarse muchos dolores de cabeza si estimula la división del proyecto en partes más manejables. En el capítulo 6 se reconsidera este tema y se plantean opciones metodológicas y administrativas para reducir el riesgo y la duración de un proyecto informático.

III Un Vistazo a la Administración de Proyectos .c1.

Una de las razones para iniciar un proyecto, con su debida administración, es que el desarrollo del sistema de información propuesto conlleva un elemento de riesgo.

En este capítulo se describen los factores que inciden en el riesgo de un proyecto informático y las técnicas administrativas apropiadas a las características del proyecto. Además, se introducen las consideraciones generales relativas a la planificación de proyectos.

1 Factores de Riesgo en un Proyecto .c2.

En [McFarlan81] se define **riesgo** como exponer el proyecto a alguno de los problemas siguientes:

- no obtener todos o algunos de los beneficios anticipados;
- costos de implementación superiores a los planeados;
- tiempo de implementación mayor al esperado;
- rendimiento técnico inferior al estimado;
- incompatibilidad con el hardware o software en uso (o planeado para el futuro).

Hay tres factores que contribuyen al riesgo de un proyecto:

- **tamaño**, que está relacionado con: dedicación del recurso humano, tiempo calendario, dinero, número de departamentos afectados y geografía;
- **tecnología**, que se mide en términos de la familiaridad que tiene el equipo del proyecto (técnicos y usuarios) con el hardware, software, la aplicación, estándares y procedimientos;

- **estructura**, que tiene que ver con la precisión usada en la definición de los objetivos y beneficios del sistema, el número y complejidad de cambios necesarios, y la actitud y grado de compromiso del usuario.

A medida que crece el tamaño del proyecto, aumenta su riesgo. Cuando la familiaridad con la tecnología o la estructura del problema son bajas, el riesgo del proyecto crece.

En el apéndice aparece un formulario para evaluar el "perfil de riesgo" de un proyecto, adaptado de [McFarlan81] y [Hammond86]. Ese mismo formulario puede utilizarse como parte de:

- la selección de proyectos (a partir del plan maestro),
- la decisión sobre la estrategia de implementación (comprar vrs. hacer, prototipos, tradicional), y
- la escogencia del estilo de administración necesario para el proyecto.

En [Hammond86] se sugiere analizar el perfil de riesgo durante las fases de evaluación inicial y de diseño arquitectónico. En la primera, el análisis del riesgo contribuye a determinar si el proyecto se aprueba, se cancela o se modifica. En la segunda, la información proveniente de la fase de especificación (análisis), permite evaluar el riesgo de las diversas opciones de diseño e implementación.

Al comparar los riesgos contra los beneficios, una empresa puede decidir cuáles tipos de proyectos desarrollará. Los proyectos con riesgos bajos y altos beneficios deben emprenderse tan a menudo como sea posible.

También pueden ser deseables aquellos proyectos con riesgos y beneficios altos, sobre todo para aquellas empresas que se encuentran en un mercado en el cual la información es determinante para obtener ventajas competitivas (finanzas, seguros, líneas aéreas). Para éstos, debe considerarse con cuidado la medida en que se aumenta el riesgo conjunto ¹² de los proyectos en desarrollo; si éste es muy alto, el nuevo proyecto deberá esperar.

Dentro del plan maestro, los proyectos con riesgo y beneficios bajos deben tener la prioridad más baja. Los proyectos de alto riesgo y bajo beneficio no deben desarrollarse (o deben reformularse).

2 Administración del Proyecto Según su Riesgo .c2.

Es una creencia generalizada que todos los proyectos se administran de la misma manera, aplicando un conjunto rígido de métodos y herramientas. Todo lo contrario, no existe una manera universalmente correcta para administrar proyectos.

Con base en las características de estructura, tecnología y tamaño, es posible seleccionar los métodos adecuados para administrar el proyecto. El perfil de riesgo del proyecto es útil para este propósito.

En [McFarlan81] se proponen cuatro tipos de herramientas, que contribuyen a la eficaz administración de un proyecto, según las características del mismo.

Las herramientas de **integración externa** incluyen los medios de comunicación que vinculan al equipo del proyecto con los usuarios (a los niveles gerenciales, medios y operativos). Entre ellas, pueden mencionarse:

- selección de un usuario como defensor del proyecto (el patrocinador);
- creación de un comité conductor integrado por usuarios;
- control de cambios manejado por el usuario;
- participación de los usuarios como miembros del equipo;
- aprobación formal de las especificaciones;
- informes de avance para el comité ejecutivo de automatización;
- responsabilización de los usuarios por la instalación y el entrenamiento;
- administración de las decisiones clave por parte del usuario.

Las herramientas de integración externa resultan significativas cuando la **estructura** del proyecto es relativamente **baja** (hay poca precisión en los objetivos, no se conoce el grado de compromiso del usuario, etc.).

Las herramientas de **integración interna** aseguran que el equipo del proyecto opera como una unidad. Estas incluyen:

- selección de un equipo técnico experimentado;
- selección del director que conduzca ese equipo;
- reuniones frecuentes del equipo;
- preparación y distribución de minutas sobre decisiones clave de diseño;
- revisiones técnicas regulares;
- baja rotación de personal en el equipo;
- selección de personas que ya han sido compañeros de trabajo, como miembros del equipo;
- participación del equipo en el establecimiento de metas y plazos;
- asistencia técnica externa - si se requiere -.

¹² Esto es, al considerar todos los proyectos en conjunto y no uno por uno.

Estas herramientas son más eficaces cuando el equipo que desarrolla el proyecto tiene una **alta familiaridad con la tecnología** que se aplicará en el proyecto (la aplicación, el hardware, el software, los estándares, etc.).

Las herramientas de **planificación formal**, que comprenden el objeto de este informe, ayudan a estructurar las tareas y recursos necesarios para completar el proyecto. Algunas de ellas son:

- redes de planificación;
- selección del modelo de ciclo de vida del proyecto y sus puntos de control;
- estándares para la especificación de sistemas;
- especificaciones para el estudio de factibilidad;
- procedimientos para aprobación de proyectos;
- evaluación post-proyecto.

La formalidad en la administración contribuye mejor al éxito de los proyectos en los cuales el equipo tiene una **baja familiaridad con la tecnología** (sea con el área de aplicación, con el sistema computacional o con los métodos para el desarrollo de sistemas).

Los procedimientos de **control formal** permiten evaluar el avance del proyecto y detectar discrepancias potenciales con el plan, de manera que puedan tomarse acciones correctivas. El control formal es más valioso cuando la familiaridad con la tecnología es baja.

3 Importancia de la Planificación y la Organización .c2.

La planificación se basa en la capacidad de visualizar los objetivos del proyecto en cuanto a su relación con las posibles limitaciones de tiempo y de costo.

La falta de planificación es la causa principal de retrasos en los proyectos informáticos, el incremento en sus costos, su baja calidad y sus altos costos de mantenimiento. Para evitar estos problemas se requiere planificar cuidadosamente el proceso de desarrollo y organizar adecuadamente al equipo de trabajo.

Se dice con frecuencia que es imposible una planificación inicial, al no conocerse con precisión los objetivos del proyecto, las necesidades del usuario y las restricciones sobre el producto. El proyecto no puede estar supeditado a la disponibilidad de información suficiente, para que se dé una planificación preliminar. Se debe reconocer que los planes preliminares se modificarán de acuerdo con la evolución del proyecto y la mejor comprensión de los requerimientos del usuario.

El plan permite "simular" actividades futuras y prever sus consecuencias. Cuando el plan goza de respaldo, se constituye en una guía para todos los involucrados. El plan es la base para controlar y coordinar las actividades del proyecto, pues facilita el seguimiento de su avance.

La planificación determina anticipadamente las actividades y minimiza la incertidumbre. La identificación de factores críticos, eventos externos y el análisis de medidas alternativas, son esenciales para una planificación eficaz.

El plan provee información sobre los costos y beneficios esperados del proyecto. Los costos pueden referirse a los del desarrollo propiamente, así como a los de operación del sistema resultante. Los administradores, al revisar el plan, pueden autorizar la continuación o cancelación del proyecto, con base en sus costos y beneficios.

El plan permite identificar necesidades de reclutamiento o entrenamiento de personal. El plan también permite evaluar el impacto del proyecto sobre: los usuarios, el hardware, el software y el departamento de informática. El plan también sirve para evaluar el impacto de los cambios suscitados, o de las modificaciones que se soliciten, sobre el sistema que se desarrolla.

4 Problemas Comunes en la Planificación .c2.

Los planes estarán expuestos a sufrir **cambios**, por lo que deberá permitirse re-planificarlos, lo cual puede resultar difícil. Afortunadamente, existen varios productos de software que facilitan algunos aspectos de la programación (tiempo, precedencia, recursos) de un proyecto.

Indudablemente, contar con tiempo suficiente para planificar tiene su **costo**. Hay una expresión muy común, que denota el error de quien busca ahorrar en pequeñeces: "no tenemos suficiente tiempo para planificar ahora; pero tendremos mucho tiempo después para corregirlo". Claramente, ésa no es una manera de ahorrar. El costo de la planificación es pequeño, comparado con el que se incurre en todo el proyecto. Bien vale la pena planificar, por los ahorros que trae consigo. Pasa lo mismo que con la programación: un poco más de tiempo en el análisis y el diseño resulta en una implementación más rápida y de mayor calidad.

Otro problema común lo provocan los **plazos impuestos** por la administración superior, motivados por problemas comerciales, legislativos u organizativos. En estos casos, lo usual es trabajar "hacia atrás", es decir, construir el plan para que se ajuste a esa situación. El resultado inevitable es un plan irrealista. Además, la actividad de planificación perderá credibilidad.

Debe recordarse que cada proyecto es **único**, de otra manera el trabajo se llevaría a cabo dentro de la organización usual. Por ello, es posible que las experiencias pasadas en trabajos similares resulten de poca ayuda para planificar todos los aspectos de un nuevo proyecto.

Igualmente difícil es **definir con anticipación el producto final**, pues éste se construye por etapas durante la ejecución del proyecto. La especificación del producto se tendrá como resultado de las etapas de especificación de requerimientos y de diseño arquitectónico. Es esencial buscar, desde el inicio, claridad en cuanto a los objetivos y alcances del proyecto.

Al contar con **personal** proveniente de diversas partes de la organización, debe tenerse en cuenta la necesidad de integrar un equipo de proyecto. Por ello, hay que dedicar tiempo y recursos para que se dé una buena comunicación entre los miembros del equipo.

5 ¿Cuándo debe Planificarse el Proyecto? .c2.

El ciclo de vida para el desarrollo de sistemas es el "reloj de alarma" para la planificación y el control de proyectos. La planificación debe iniciarse después de la evaluación inicial (en la etapa que aquí se denomina "planificación"), y busca construir un plan preliminar.

Es preferible que la planificación no se extienda como un proceso continuo a lo largo del proyecto, sino cuando ocurre alguno de estos tipos de evento:

- al inicio de cada fase del ciclo de vida, después de una revisión de los resultados parciales producidos por la fase precedente;
- de manera *ad-hoc*, cuando ocurre algún cambio o problema importante, que amerita una revisión de los planes previos.

Cuando se inicia una fase del proyecto conviene hacer una revisión del plan original, con el fin de ajustarlo a la realidad que ha revelado el trabajo hecho en las fase previas. Se pone especial cuidado en detallar las tareas de la fase que está por iniciarse, así como a la asignación de recursos en la nueva fase (especialmente los recursos humanos).

Los dos momentos en que se hacen ejercicios de planificación más profundos son:

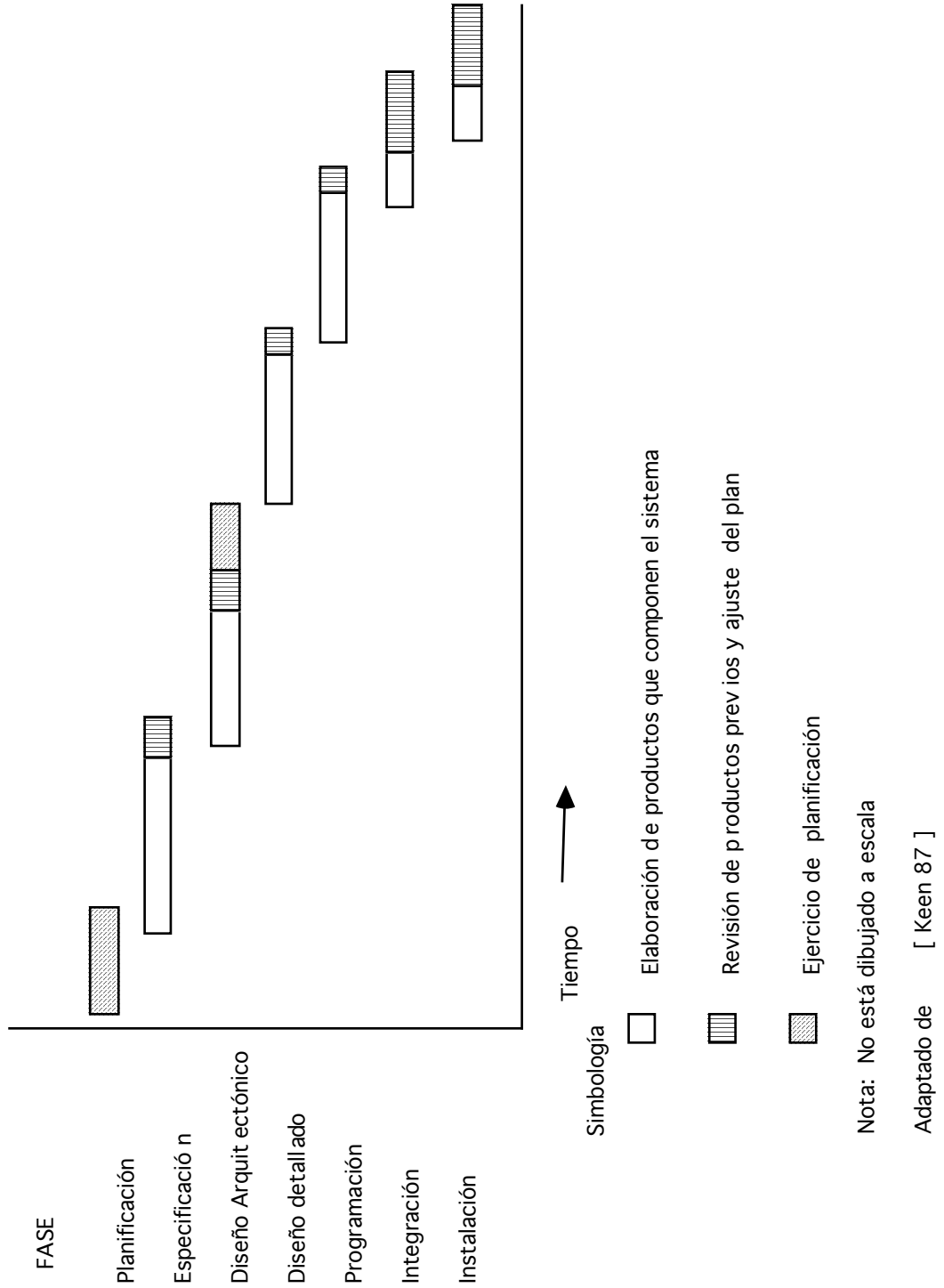
- en lo que aquí se denomina la fase de planificación, es decir, una vez que se ha decidido desarrollar el sistema de información y se ha designado el director del proyecto;

- antes de iniciar el diseño detallado, pues en ese momento el diseño arquitectónico brinda una visión general de las características del sistema, así como información para decidir sobre las estrategias adecuadas para su construcción.

Conviene tener en cuenta lo señalado arriba, con el fin de prever la asignación de suficientes recursos (humanos y tiempo) durante esos "picos" de planificación.

La figura 2 muestra los momentos de planificación previsible, de acuerdo con el ciclo de vida:

Figura 2 La planificación en el desarrollo de sistemas.



6 ¿Quiénes Planifican el Proyecto?.c2.

La planificación del proyecto es conducida por el director del proyecto, con diversos grados de énfasis según la fase del proyecto que esté por iniciar.

El director tendrá la responsabilidad general del plan. Él velará por la determinación de los posibles beneficios del proyecto, así como de los costos que se atribuyan al usuario. Para esta cuantificación de beneficios y costos son importantes:

- el respaldo del patrocinador, y
- la habilidad del director de proyecto para coordinar su trabajo con el del los jefes de las áreas usuarias.

El director deberá consultar con el jefe de desarrollo de sistemas (o su equivalente) los aspectos relacionados con el personal que estará a su cargo.

El director recopilará información técnica relativa a estimaciones, costos, itinerarios, estándares, etc. Con el fin de tener mayor claridad sobre el trabajo que se planifica, el director revisará la documentación de algunos de los proyectos ya terminados. En particular, estará interesado en los planes, el legado del proyecto ¹³ y la evaluación post-proyecto ¹⁴.

13 El legado es un documento que resume el proyecto (en cuanto a lo que se esperaba, lo que se logró y lo que faltó) y da cuenta de la historia del proyecto (dedicación de esfuerzo, aciertos, errores). Un buen legado da recomendaciones para proyectos futuros.

14 Estas evaluaciones son convenientes unos meses después de terminado el proyecto y entregado el producto (sistema). Ayudan a poner el proyecto en perspectiva, de acuerdo con los resultados obtenidos a partir de la experiencia con el sistema.

Para el éxito del plan, es importante que el director lo someta a la consideración de los integrantes (técnicos) del equipo de proyecto. Esta puede ser una manera de asegurar que no se han descuidado aspectos, así como de lograr que el equipo se identifique con el proyecto desde etapas muy tempranas. La participación de los miembros del equipo en la planificación es un factor de motivación, así como una oportunidad para comenzar a integrarse como grupo.

El comité conductor (o, en su ausencia, el patrocinador) revisará el plan del proyecto presentado por el director, para decidir sobre la asignación de recursos, con base en los costos y beneficios del proyecto. Para los proyectos de gran envergadura, es recomendable que el comité ejecutivo de automatización revise el plan.

7 Notas Bibliográficas .c2.

El lector interesado sobre otros aspectos de la administración de proyectos informáticos puede consultar: [Keen87], [Page-Jones85], [Allen82], [Rubin86], [Nolan76], [DeMarco82] y [Trejos84].

[Keen87] y [Page-Jones85] ofrecen una amplia discusión sobre la generalidad de los aspectos que comprenden la administración de proyectos informáticos. Ambos se basan en la amplia experiencia empresarial de sus autores como directores de proyecto y consultores (en empresas de países desarrollados).

[Bruce82] es también un tratado sobre administración de proyectos, pero está muy orientado a trabajos contratados por el Departamento de Defensa de los Estados Unidos; aún así, contiene excelentes guías sobre la documentación y presentación de proyectos. [Gilb88] incluye enfoques novedosos sobre administración de proyectos, que resultan muy

distintos de los enfoques más populares sobre el tema.

[McFarlan81] y [Hammond86] ofrecen discusiones más detalladas sobre el tema de la evaluación del riesgo y la selección de las formas de administración adecuadas a las características de cada proyecto.

En [Thayer80] y [Thayer82] se estudian varios problemas peculiares de la administración de proyectos de ingeniería de software, así como algunas de las mejores soluciones que se les han encontrado, según una encuesta aplicada a profesionales que laboran en empresas y centros de investigación de los Estados Unidos.

En [Head82] y [Head84] se pone a la administración de proyectos dentro del contexto de la planificación estratégica de los sistemas de información en una empresa.

IV Organización del Equipo del Proyecto .c1.

Los proyectos tienen una vida finita. En contraste, una empresa, gobierno u otra organización, espera existir indefinidamente. Esta diferencia **temporal** plantea dificultades específicas a la organización y administración de un proyecto dentro de una entidad organizacional mayor, como lo es una empresa.

Además, los proyectos requieren frecuentemente la utilización parcial de recursos, mientras que la empresa - que es permanente - trata de usar sus recursos a tiempo completo.

Aunque ninguna forma de organización es perfecta, debe reconocerse la existencia de proyectos cuando están presentes. Esto significa que la empresa debe planear cómo alojar este disturbio temporal y aceptar alguna falta de armonía en su forma acostumbrada de proceder.

1 Opciones de Organización .c2.

Existen varias maneras en las cuales las empresas, sus divisiones, o las entidades gubernamentales pueden organizarse para administrar proyectos. Las formas más comunes son: funcional, proyecto y matriz ¹⁵.

1.1 Organización Funcional .c3.

Esta es la forma de organización más antigua. Se la usa poco en organizaciones orientadas al desarrollo de sistemas o de otros productos. Es usual que el proyecto se divida en segmentos y que estos se asignen a las áreas funcionales relevantes. El proyecto es coordinado por los niveles administrativos funcionales o superiores.

Su principal problema es que todas las decisiones que cruzan una frontera funcional son hechas por un individuo, el director del área funcional. Muy pocos administradores pueden lidiar con tanta autoridad, por lo que son frecuentes los cuellos de botella, que provocan: retrasos, gastos excesivos o baja calidad.

Las ventajas de una organización funcional son:

- permite ejercer control estrecho y centralizado;
- aprovecha factores comunes a varios proyectos;
- asigna personal al establecimiento de estándares y metodologías avanzadas de desarrollo;
- reduce la deserción de personal al terminar el proyecto;
- se adapta a los requerimientos de largo plazo;
- se centra en el individuo y no en el proyecto.

Sus desventajas son:

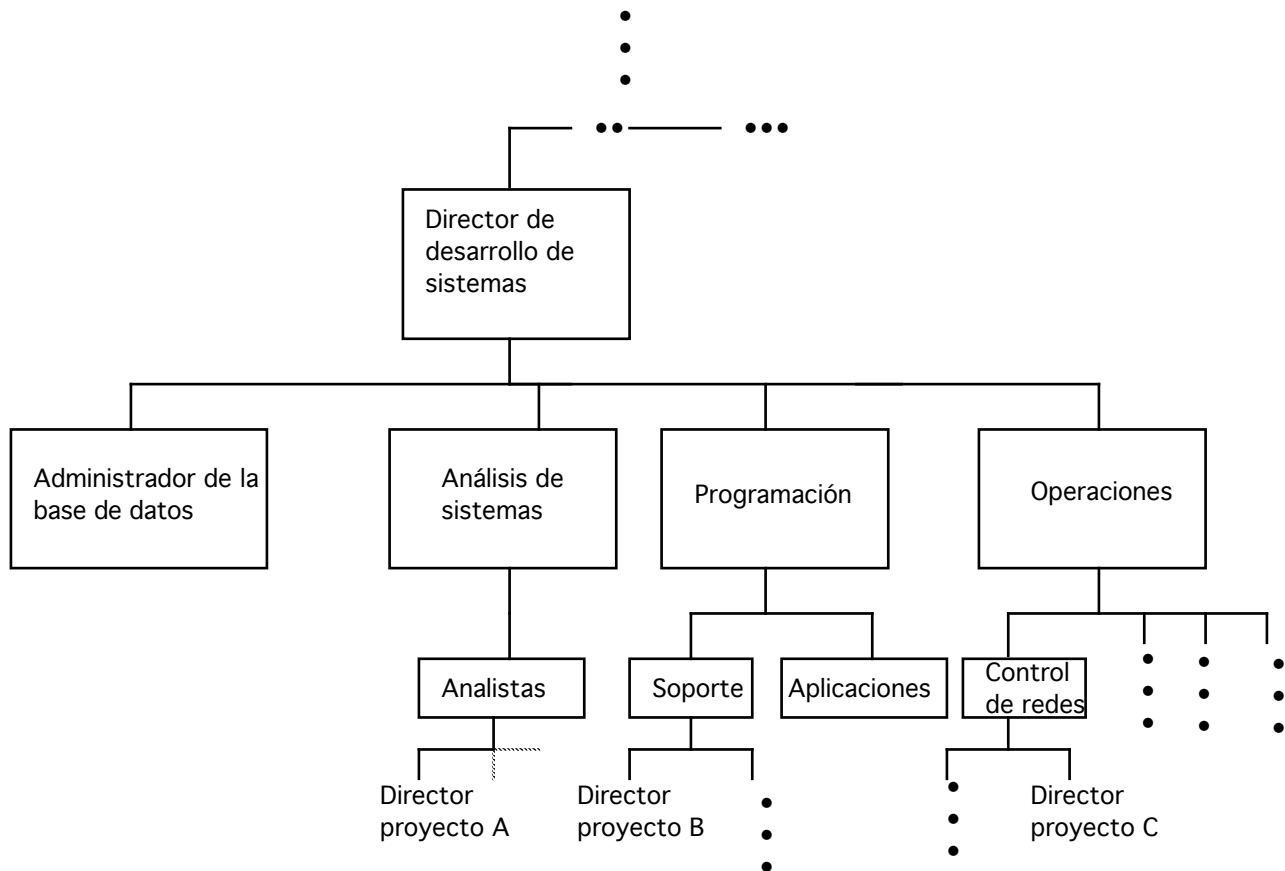
- la resolución de problemas entre una área funcional y el proyecto la hace un individuo;
- limita la formación de generalistas en sistemas, al favorecer la especialización funcional;
- el control de proyectos es pobre, en términos de tiempo, costo y calidad;
- puede degenerar en un estilo de administración burocrático y en "*ad-hocracia*" ¹⁶.

La figura 3 muestra una organización funcional típica.

¹⁵ El tratamiento que se dará a este tópico será un tanto teórico.

¹⁶ Es decir, en una administración coyuntural, o gobernada por las circunstancias.

Figura 3
Organización funcional típica



1.2 Organización por Proyecto.c3.

En el extremo opuesto a la organización funcional se encuentra la organización por proyecto. En este caso, se requiere que la mayoría de los recursos (al menos un 70%) necesarios para completar el proyecto se encuentren bajo la autoridad de un solo administrador. Todo el personal de tiempo completo se asigna formalmente al proyecto. Los administradores de las áreas funcionales no tienen involucración directa dentro del proyecto, aunque pueden - por supuesto - influir su curso.

Las ventajas de la organización por proyecto son:

- las decisiones técnicas y administrativas son rápidas y se toman en un nivel organizacional bajo. Se ejerce un mejor control.
- las interfaces entre áreas funcionales son mínimas. La responsabilidad se define estrictamente.
- permite formar generalistas en sistemas y en administración de proyectos.
- la motivación del personal y su identificación con el proyecto son altas durante el período activo de desarrollo del sistema.

Sus desventajas son:

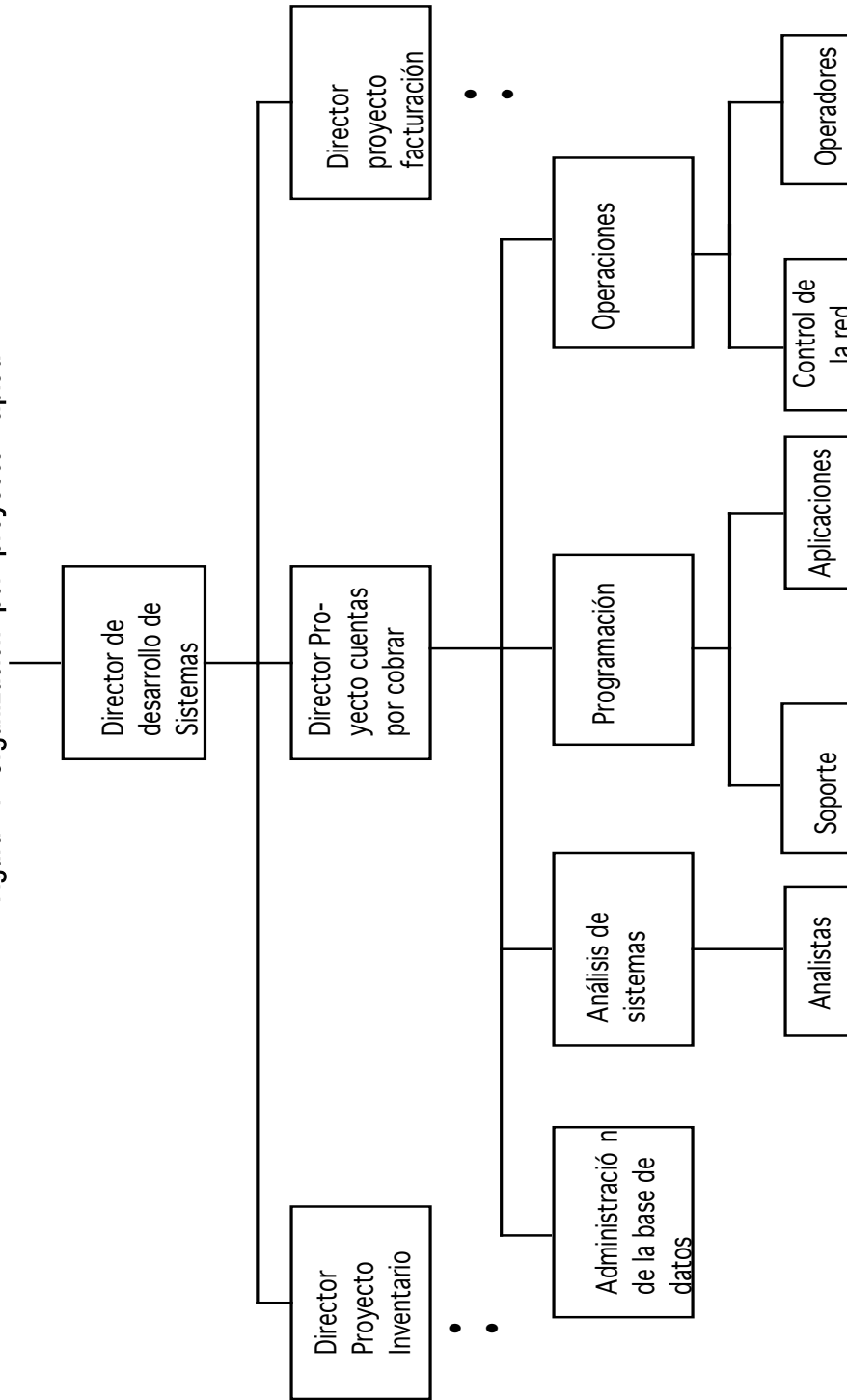
- los proyectos deben ser pequeños;

- es posible que la alta administración pierda la pista del avance del proyecto, si no se incluyen las previsiones administrativas para mantener el control ¹⁷;
- no pueden alcanzarse economías de escala para algunos recursos críticos (los especialistas);
- entrenamiento costoso de especialistas para las diversas fases del desarrollo;
- es difícil mover programadores entre proyectos, pues cada programador se asigna exclusivamente a un proyecto;
- se da mucha deserción al final del proyecto;
- se inhibe el aprovechamiento de factores comunes entre proyectos y la generación de buenos estándares para el desarrollo de sistemas.

La figura 4 muestra una organización por proyecto típica.

¹⁷ Por ejemplo, mediante comités conductores.

Figura 4 Organización por proyecto típica



1.3 Organización Matricial .c3.

Las organizaciones funcional y por proyecto tratan de optimizar alguna restricción organizacional:

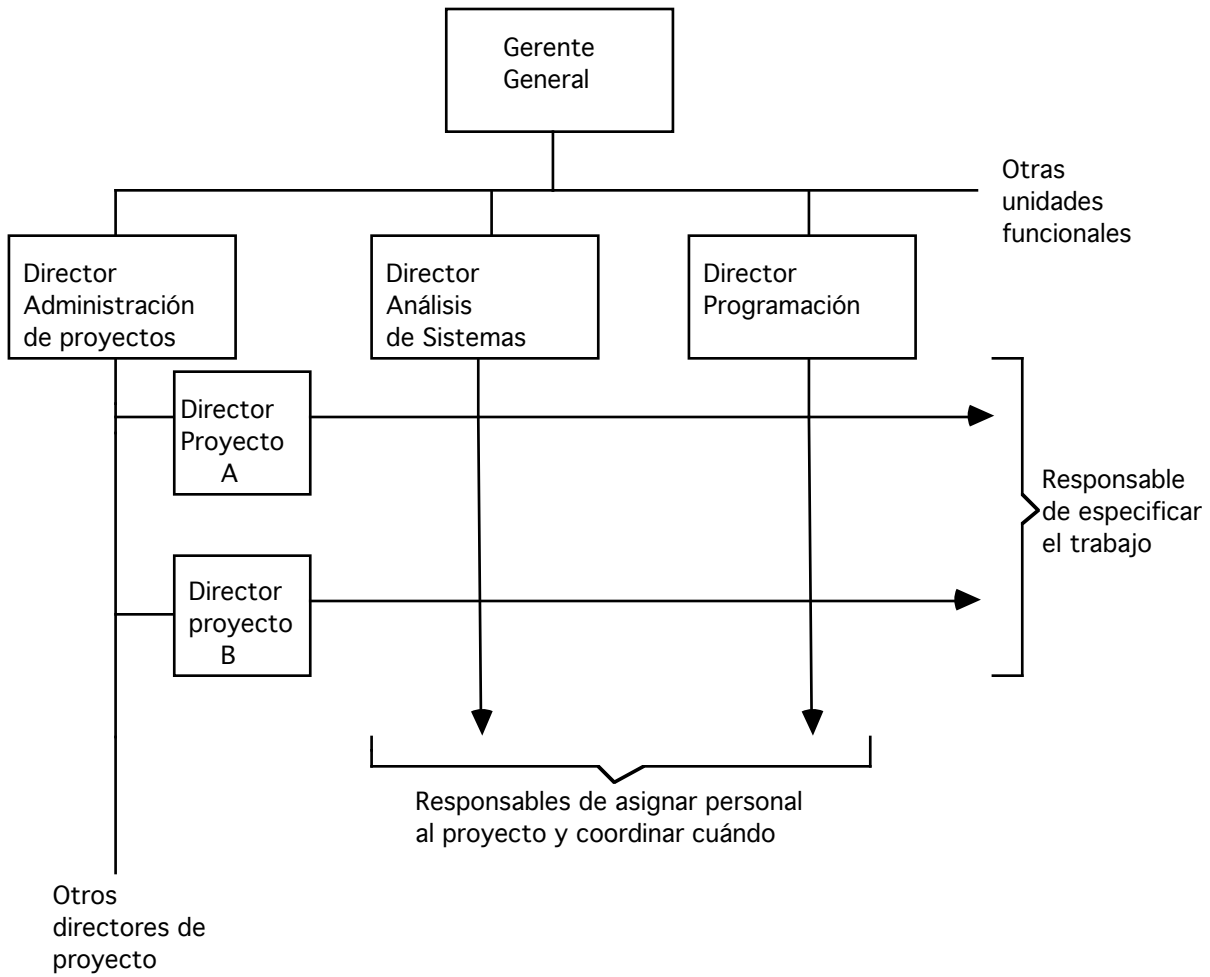
- la funcional se concentra en las **metas de largo plazo** (factores comunes, avance tecnológico, mejores estándares, economías de escala),
- la organización por proyecto lo hace en las **metas de corto plazo** (calendario, costo, calidad).

La organización matricial es una estructura compleja que trata de optimizar ambas restricciones simultáneamente. Constituye, en esencia, un compromiso entre la estructura funcional tradicional y la organización por proyecto pura. Es más flexible que la funcional, pero no tanto como la organización por proyecto.

La matriz es más eficiente que organización por proyecto, pero incurre en costos administrativos que son innecesarios en la organización funcional. Quienes la han usado estiman que requiere una administración cuidadosa, especialmente en cuanto a su doble línea de mando.

La figura 5 ilustra una organización matricial.

Figura 5 Organización matricial Típica



En general, sus ventajas son:

- hace un uso eficiente de los recursos; pueden compartirse especialistas y equipo entre varios proyectos;
- existe un mecanismo claro y factible para coordinar el trabajo entre unidades funcionales;
- la comunicación vertical y horizontal se mejora;
- el contacto frecuente entre miembros de unidades diferentes permite respuestas que logran adaptarse al cambio;
- los expertos funcionales se mantienen juntos, a pesar de la ida y venida de proyectos;
- la participación de los miembros del equipo en la toma de decisiones aumenta su motivación y compromiso.

Sus desventajas son:

- hay luchas de poder al traslaparse la autoridad y la responsabilidad entre el proyecto y las áreas funcionales;
- hay mayor conflicto al competir por recursos escasos (especialmente el personal);
- la toma de decisiones puede atrasarse, ya que se hacen más comunes las consultas;
- resulta difícil evaluar el avance del proyecto, dada su constitución multidisciplinaria;
- tener directores de proyectos conlleva costos adicionales;
- las tensiones aumentan, al tener que rendir cuentas a dos jefes, lo que provoca ambigüedad y conflicto de papeles.

Hay tres variantes de la estructura matricial, que dependen de la influencia que tienen, sobre el proyecto, los administradores de las áreas funcionales y el director del proyecto.

Una **matriz funcional** ocurre cuando el director del proyecto se limita a planificar y coordinar los esfuerzos de los grupos funcionales involucrados. Los administradores funcionales son responsables por el diseño y finalización de los requerimientos técnicos dentro de su área o disciplina.

Una **matriz de proyecto** se refiere a la situación en la cual el director del proyecto es responsable de la terminación del proyecto y tiene autoridad directa sobre el personal y su flujo de trabajo. Los administradores de las áreas funcionales asignan el personal según se requiera, y proveen servicios de apoyo y asesoría experta.

Una **matriz balanceada** es aquella en la cual el director del proyecto se hace cargo de definir **qué** necesita alcanzarse (planificar, integrar las contribuciones de los grupos funcionales, controlar el avance). Los administradores de las áreas funcionales se responsabilizan por el **cómo**, es decir, asignar personal y ejecutar su segmento del proyecto de acuerdo con los estándares y calendario establecidos por el director del proyecto. El director del proyecto coordina la verificación de las contribuciones de los grupos funcionales, contra las especificaciones que él proporcionó.

Larson y Gobeli (ver [Larson87]) estudiaron en detalle el uso de estas variantes, así como su evaluación, por parte de 510 administradores de proyectos o gerentes de empresas que ejecutan proyectos. Un 77% de las empresas han usado la organización matricial y un 89% de ellas (69% del total) están dispuestas a usarla de nuevo. La variante que fue considerada más eficaz fue la matriz de proyecto.

Pasar de una organización funcional tradicional a una matriz de proyecto puede ser un cambio demasiado radical. Al implementar una organización matricial se deberá vencer una alta resistencia al cambio y más de un obstáculo político.

Para una empresa que jamás ha experimentado con la organización matricial para llevar a cabo sus proyectos, es recomendable que comience con la matriz funcional, evolucionando luego a la matriz balanceada, para terminar en la matriz de proyecto.

Este proceso de cambio debe ser gradual, con el fin de que la empresa interiorice la forma de organizar proyectos, hasta que llegue a ser parte de su "cultura organizacional".

2 Metas de la Organización .c2.

Además de la estructura organizativa escogida para el equipo del proyecto, deben fijarse metas que conviene alcanzar.

Debe lograrse **concordar la pericia de un trabajador con el trabajo que realiza**. La manera de lograrlo es hacer un plan en el cual se detallan las actividades por realizar y las habilidades requeridas para llevarlas a cabo. Con base en estas habilidades, el director buscará al personal idóneo.

No siempre se puede encontrar personal que reúna todos los requisitos. En estos casos, pueden reclutarse varias personas cuyas habilidades sean complementarias, capacitar personal existente, o contratar personal externo.

En cualquier caso, debe **evitarse** crear tareas con el único propósito de hacerlas calzar a la gente disponible.

Con base en las cualidades del personal que se ha podido reclutar para el proyecto, deben revisarse las estimaciones de tiempo hechas.

Hay que **lograr el estilo de trabajo deseado**. Esto significa escoger entre un enfoque de trabajo con generalistas o con especialistas.

Un estilo generalista permite que el equipo del proyecto sea bastante estable. Los problemas de comunicación entre fases se reducen y los miembros pueden trabajar independientemente unos de otros.

En la realidad, es casi imposible encontrar gente con conocimientos y experiencia a la vez amplios y profundos. Es difícil mantener la consistencia en todo el proyecto y los errores se evidencian mucho más tarde. La realización del trabajo es lenta, al pasar el trabajador de una fase a otra. Un enfoque totalmente generalista no tiene sentido sino para los proyectos más pequeños.

Para la mayoría de los proyectos, el estilo especialista es el único realizable. Sus ventajas son:

- los trabajadores pueden aplicar su conocimiento en profundidad;
- los problemas de consistencia se reducen;
- cada persona puede trabajar al máximo de eficiencia, al no cambiar el marco de referencia al cual está habituado;
- los errores se descubren al pasar de una fase a la siguiente.

La tercera meta es **minimizar el número de trabajadores ociosos**. La pereza es la madre de todos los vicios. Si no hay trabajo por hacer, es preferible que un trabajador retorne a su área funcional (suponiendo una estructura matricial), a que su moral decaiga por no tener nada significativo que hacer.

Si la pugna por obtener personal es muy fuerte, deberá asignarse al miembro del equipo - por el cual se está compitiendo - una genuina tarea del proyecto, aunque sea alguien que no reúne todas las habilidades requeridas. En este caso, hay que darle también oportunidad de aprender esas habilidades y conocimientos. También será imprescindible una explicación del porqué se le está dando esta aparentemente extraña asignación.

El director deberá tener claro que, para una especialidad determinada, la productividad de un miembro no-experto es inferior a la de un experto.

Para obtener la máxima productividad **debe asignarse una sola tarea a la vez a cada trabajador**. De otra manera, el trabajador tendrá que cambiar de contexto continuamente y esperar (o adivinar) una priorización de las actividades a su cargo.

Cuando se tienen más actividades por realizar concurrentemente que personal disponible, la única solución es contratar personal temporal. Muchas experiencias muestran que resulta más barato tener un equipo humano relativamente pequeño - pero manejable -, que trabajar con uno multitudinario.

3 Evolución de la Organización Durante el Proyecto .c2.

Cada fase del proyecto requiere de su propia mezcla de habilidades y de un ajuste en la organización. Por esta razón, al personal del proyecto debe aclarársele que su puesto es temporal.

Al inicio del proyecto será común tener muchos analistas y diseñadores de sistemas, contra pocos programadores. Hacia la mitad del proyecto, habrá muchos programadores y menos analistas. Durante la integración de subsistemas, los

analistas y diseñadores vuelven a tener un papel importante.

Asimismo, la participación del usuario será mayor en las fases de especificación de requerimientos, diseño arquitectónico, aceptación e instalación.

Al convertir bases de datos o iniciar la captura de datos en la fase de instalación, deberá preverse la participación de digitadores y otro personal con las cualidades necesarias.

La figura 6a ilustra una distribución del personal especializado, en el curso de un proyecto típico. La figura 6b muestra los grados de participación de los usuarios, la administración superior y la auditoría (o control de calidad) a lo largo del proyecto.

Figura 6 a. Niveles de participación técnicos del equipo del proyecto.

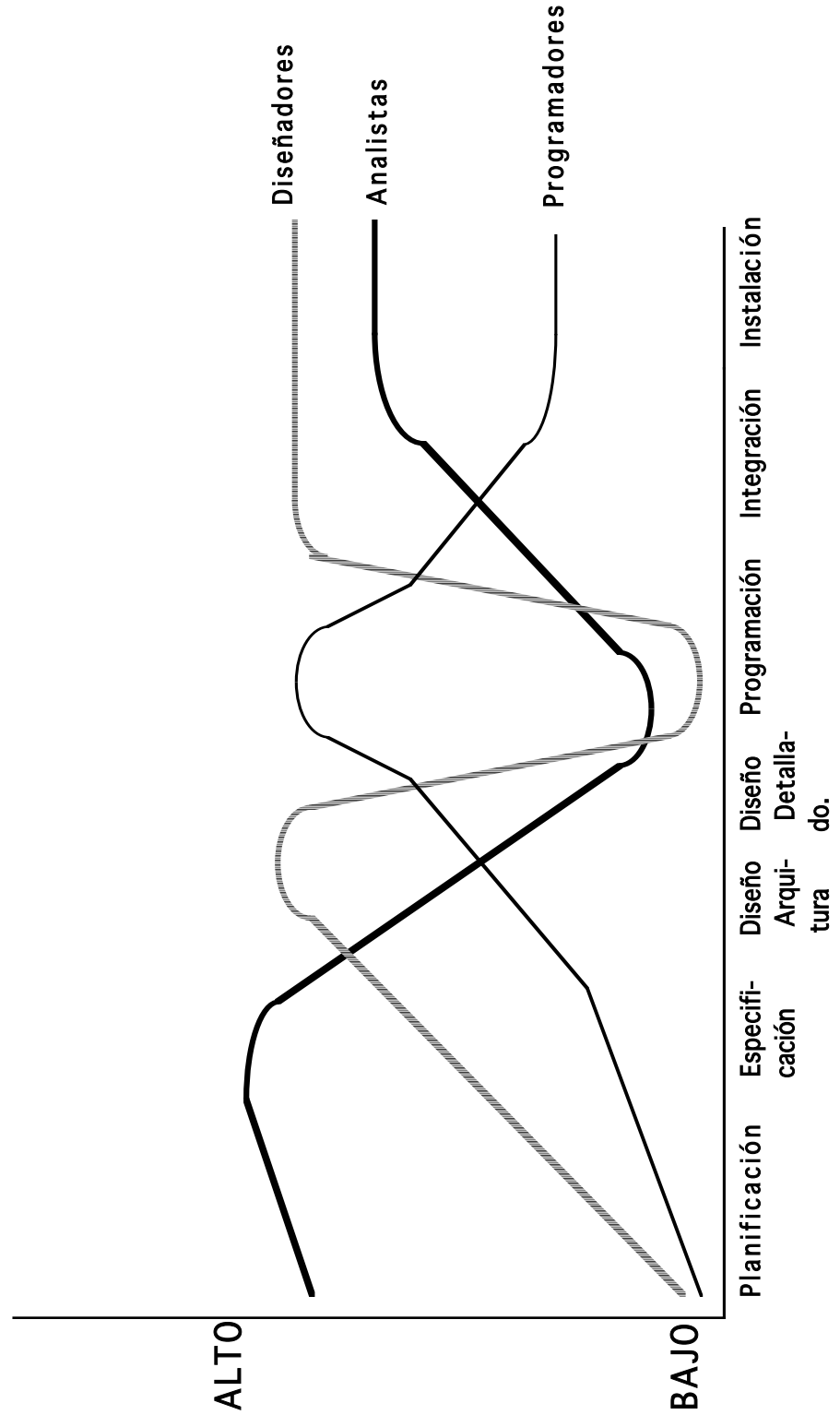
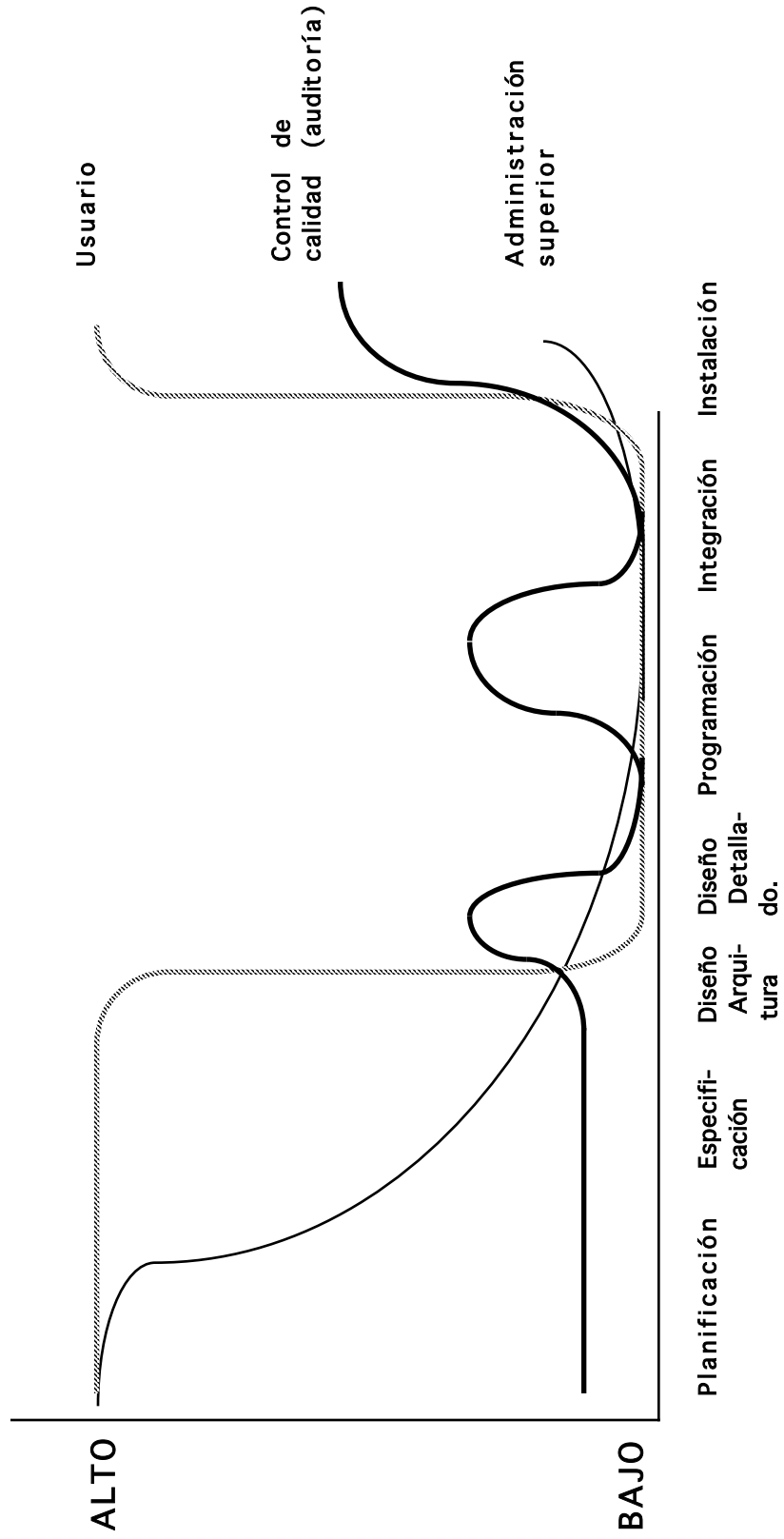


Figura 6 b Niveles de participación usuarios, administración, auditoría.



V Aspectos Técnicos de la Planificación .c1.

El plan del proyecto es un documento que describe cómo se supone conducir el proyecto que desarrollará el sistema de información. El plan deberá definir las tareas y los recursos requeridos para alcanzar a tiempo los objetivos establecidos. Sin importar cuán pequeño sea el proyecto, es necesario un plan que defina **qué** debe lograrse, por **quién** y **cuándo** se realizará, y que indique **cuánto** costará.

El plan no tiene por qué ser demasiado complicado, pero debe definir con precisión cuál es el trabajo por el que el director del proyecto será responsable. También se definirán los métodos que garanticen que el trabajo se realizará exitosamente. El plan es el documento de referencia para informar acerca del progreso del proyecto, a lo largo de su ciclo de vida.

En este capítulo se discutirá sobre algunas técnicas que se usan para planificar los proyectos. Las técnicas se basan en algunas premisas administrativas ([Murdick77]):

- todo trabajo puede ser planificado y controlado;
- cuanto mayor es la dificultad para planificar el trabajo, mayor es la necesidad de planificar;
- la asignación de la administración del proyecto a un director con amplias responsabilidades es un factor importante para incrementar las probabilidades de éxito de un proyecto.

El director del proyecto debe controlar todos los costos del proyecto. Sin embargo, debe dirigir el proyecto, aunque no tenga mando directo sobre todas las personas que participan en él. Si se cuenta con una estructura del desglose del trabajo se tendrá una clara definición de la descomposición del proyecto y la asignación de responsabilidades.

1 Estructura del Desglose del Trabajo y Asignación de Tareas .c2.

A partir del modelo de ciclo de vida escogido para el proyecto, las características del sistema por desarrollar y de la organización elegida, se realiza una descomposición de tareas, hasta llegar a unidades que pueden asignarse a los individuos que conforman el equipo del proyecto.

La **estructura del desglose del trabajo** (EDT) permite hacer evidente esa descomposición jerárquica y la asignación de tareas a los miembros del equipo. La EDT permite identificar la integración de resultados parciales. La EDT es el primer ejercicio que hace el planificador para comprender la naturaleza del proyecto que se desarrollará.

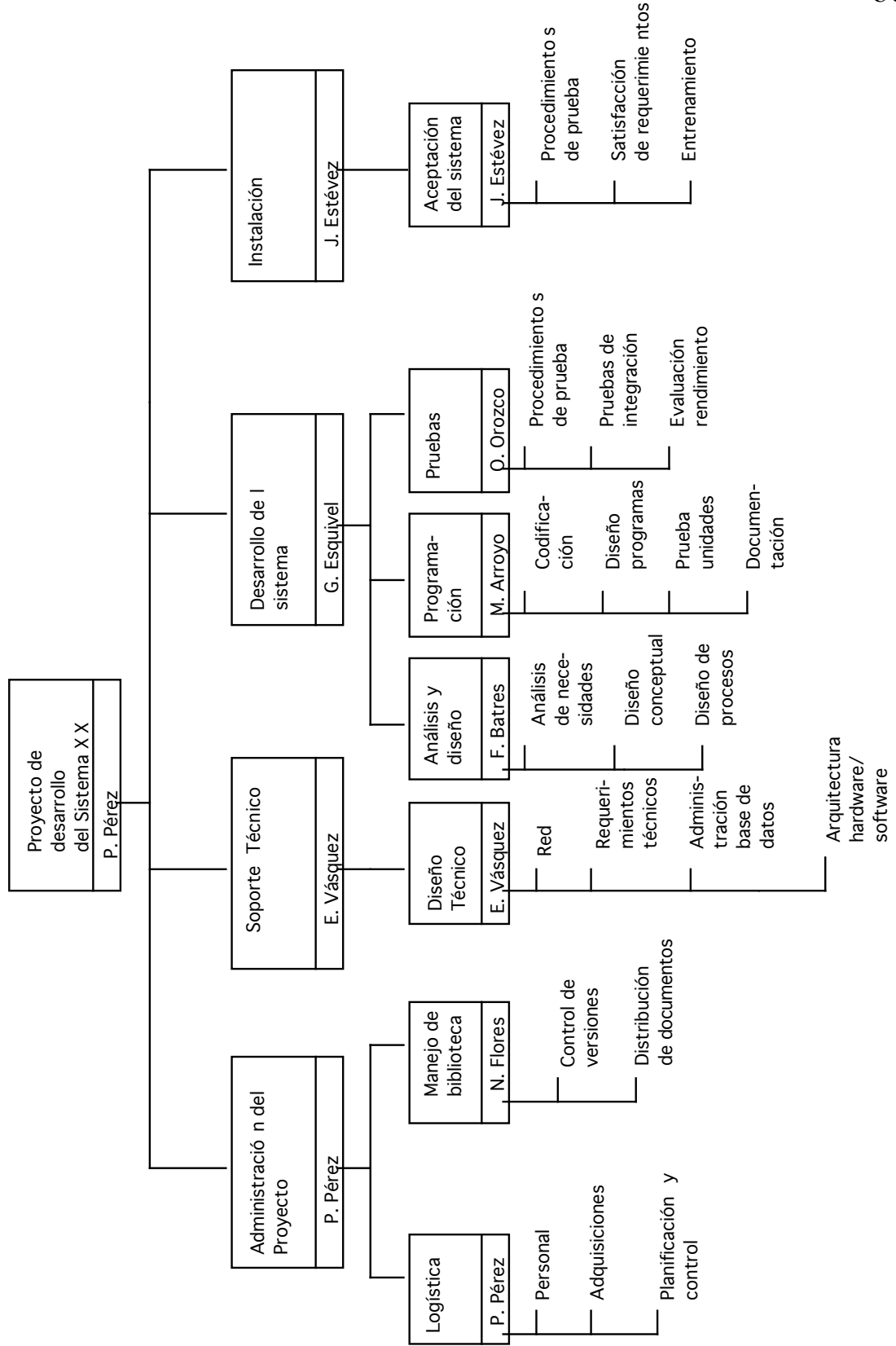
La EDT puede hacerse por productos o por procesos. La EDT comienza con el resultado final deseado y luego se descompone jerárquicamente en tareas. Se definen las tareas más importantes, que continúan subdividiéndose hasta llegar a los **paquetes de trabajo**, que son los elementos más pequeños de la EDT.

Los paquetes de trabajo establecen el grado de visibilidad y control administrativo que se ejercerá sobre el proyecto. Las tareas deben ser de un tamaño tal que haga manejable el trabajo a quien se le asigna, así como permitir al director dar seguimiento al proyecto. La duración de una tarea no debe exceder las tres semanas, pues más allá de este plazo se pierde el control con facilidad.

Cada una de estas tareas debe llegar a producir un resultado tangible, de manera que, para efectos de control, la tarea sólo puede adoptar uno de tres estados posibles: "terminado", "no iniciado" o "incompleto".

Los paquetes de trabajo pueden constituir una unidad básica para la determinación de costos (ver sección 5.4). En proyectos complejos, a cada paquete se le puede asignar un número de cuenta, a fin de llevar una adecuada contabilización de los costos. En la figura 7 se muestra una EDT idealizada.

Figura 7 Ejemplo de estructura de desglose del trabajo.



La EDT provee la base para la estimación de costos, la organización del proyecto, la calendarización de actividades y el seguimiento de los costos. Al preparar la EDT se recalcan las actividades de **integración**. Los puntos de unión frecuentemente implican que una actividad de verificación debe realizarse, lo que ayuda a identificar otras necesidades de asignación de recursos. Es también importante incluir, dentro de la EDT, actividades como preparación de informes, revisiones y coordinación.

Al momento de decidir si se justifica el proyecto, es recomendable sacar del plan la etapa de mantenimiento, aunque se tomen en cuenta sus posibles costos. No es conveniente copiar a ciegas la EDT de un proyecto anterior, aunque sí confrontar la EDT contra listas de comprobación compiladas previamente, con el propósito de corroborar si está completa. Lo importante es preparar concienzudamente la EDT, con el fin de aumentar las probabilidades de éxito del proyecto.

El director debe poner especial cuidado en planificar actividades que involucran al comité conductor (o patrocinador + gerente de información + usuarios, en su defecto), típicamente en lo que concierne a las **revisiones** del avance, así como la aprobación de los resultados parciales y finales del proyecto.

Si es posible, el director debe buscar que otra persona revise su EDT, o bien, pedirle que le haga una EDT general (hasta el tercero o cuarto nivel). Esto ayudará a descubrir discrepancias o descuidos en la construcción de la EDT, para así reducir los problemas futuros en el proyecto.

Cuando se ha hecho la EDT inicial, la planificación del itinerario (calendario) puede comenzar.

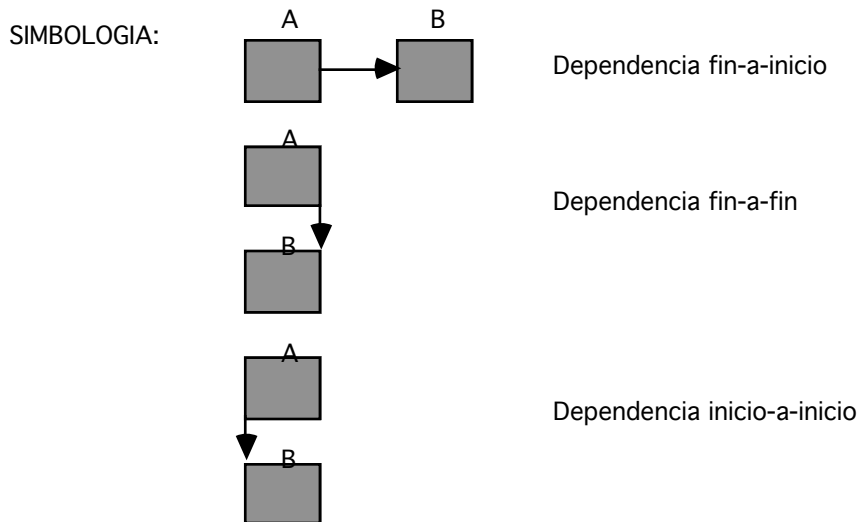
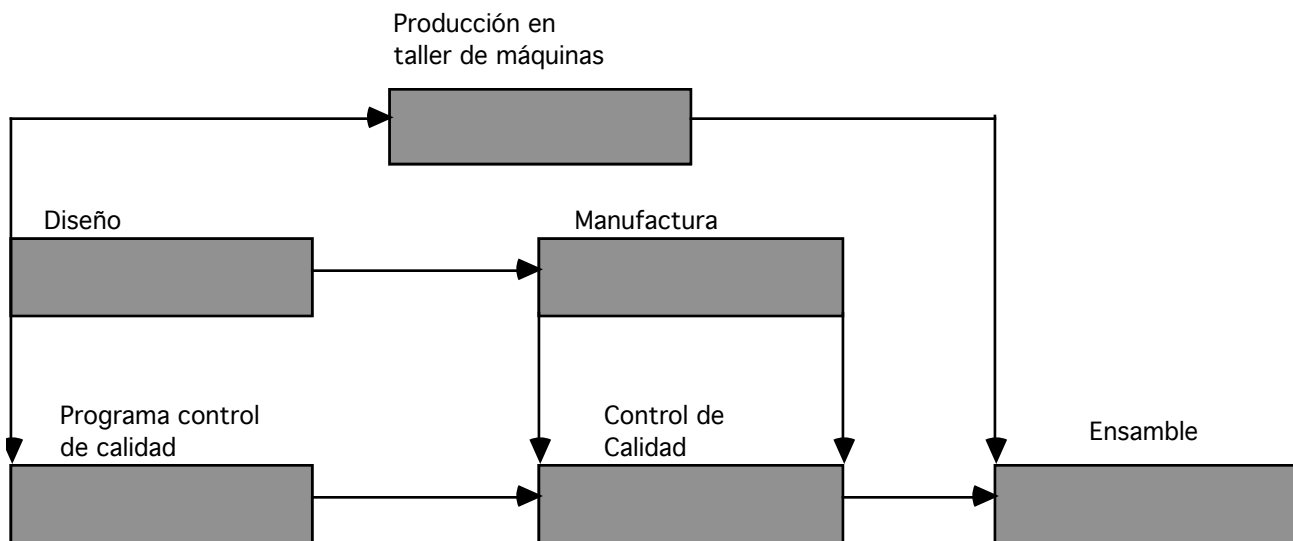
2 Calendarización (Itinerarización) .c2.

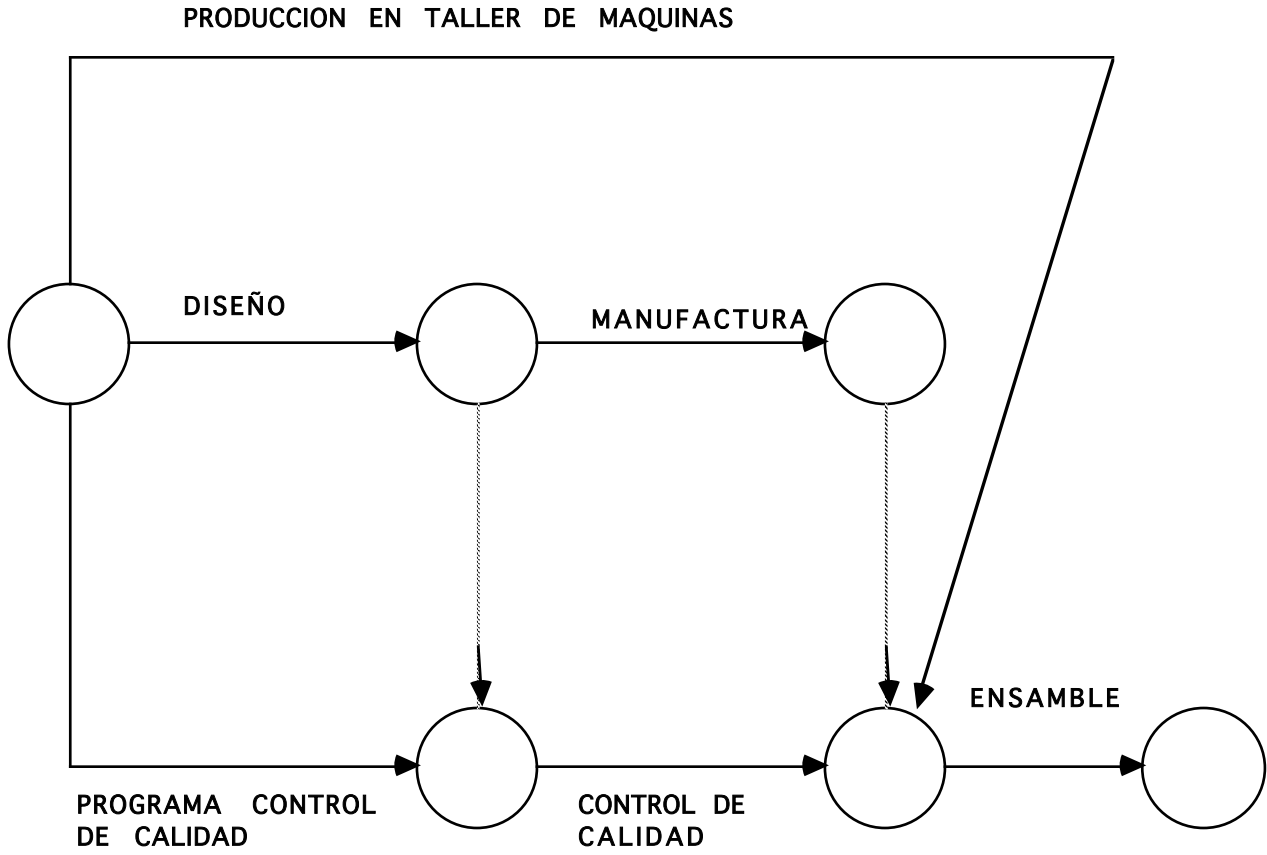
Las primeras herramientas que se usaron para calendarizar fueron las listas de comprobación y los gráficos de Gantt (o de barras). Aunque estas últimas pueden servir para visualizar la duración planeada del proyecto y de su avance, no se muestran las dependencias entre las actividades con claridad.

En contraste, las técnicas basadas en **redes**, como PERT y CPM, resuelven el problema de mostrar la dependencia entre las actividades del proyecto y permiten, de manera sencilla, montar la planificación de tiempos, recursos y costos. Además, hay un sinnúmero de paquetes de software disponibles para facilitar el trabajo con redes de planificación, que corren en prácticamente cualquier tamaño de computadora. El uso de redes de planificación resulta valioso a medida que crece la complejidad de los proyectos; para las redes más complejas usar un paquete de software resulta prácticamente imprescindible.

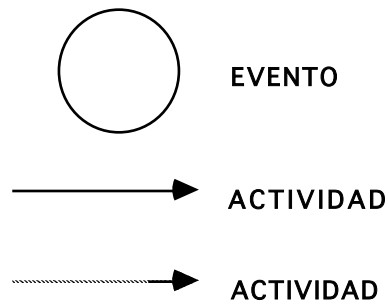
Hay dos maneras básicas de representar gráficamente la red: **redes de actividades sobre nodos** y **redes de flechas**. En las figuras 8a y 8b se muestra la misma red, representada usando ambas convenciones.

Figura 8a Red con actividades sobre nodo.





SIMBOLOGIA



Como se ve, la diferencia es la representación gráfica (y algunas reglas para calcular la **ruta**)

crítica¹⁸. La experiencia ha demostrado que la representación de actividades sobre nodos resulta más fácil de aprender.

El primer paso para construir la red es comenzar con la lista de actividades que se obtuvo en la EDT. Seguidamente, se diseñará la **red lógica**, que provee una ilustración gráfica de las diversas actividades y de las dependencias entre ellas. Con base en las estimaciones de recursos y tiempos (ver sección 5.4), se definirá la red final.

Algunos directores de proyectos se resisten a utilizar las redes de planificación porque las consideran muy complejas, o porque no tienen un paquete de computadora que los asista. Esto es un error, ¡porque lo que es complejo es el proyecto! En efecto, si no se puede dibujar una red para un proyecto, es una señal de alarma: no se entiende el proyecto.

Otro problema es el nivel de detalle de la red, el cual debe ser adecuado al nivel administrativo que le dará consideración. Además, demasiado detalle en el itinerario dificultará su control eficaz. Resulta conveniente jerarquizar la red, al agrupar varias actividades relacionadas en una sola y mantener sub-redes para ellas.

Debe recordarse que la red es parte del plan y será un importante instrumento de control sobre el avance del proyecto. La red permitirá comparar lo planeado contra lo realizado y, con base en eso, informar a la administración acerca del avance del proyecto. Eventualmente, se tomarán acciones correctivas, si la situación lo amerita.

Cuando se diseña la red, conviene pensar en el procedimiento por seguir para informar a la administración. La experiencia sugiere que el grado de formalidad de los informes sea proporcional al tamaño del proyecto. Si el procedimiento es excesivamente formal, en el caso de los proyectos pequeños se corre el riesgo de entorpecer las acciones correctivas sobre el proyecto, que habrán de aplicarse en caso de desviarse de lo planeado.

Los paquetes de software se encargan de calcular la ruta crítica, balancear la carga de trabajo entre los recursos disponibles (por lo común escasos) y brindar información para facilitar el control sobre el proyecto.

Aunque los paquetes para calendarización de proyectos son muy populares, el director del proyecto debe tener presente no caer en la **trampa del "95% completo"**, mencionada en el capítulo 1.

Aún para los proyectos más pequeños, conviene usar un paquete de calendarización. Es preferible que éste sea fácil de usar (MacProject, por ejemplo). Existen paquetes más capaces, pero su uso es más complicado; deben usarse cuando no hay otra opción.

3 Estimación de Recursos .c2.

Uno de los aspectos más difíciles de la planificación de proyectos es la estimación del tiempo que se requiere para desarrollar el sistema de información. Las estimaciones son una aproximación de los días (semanas, meses) de esfuerzo necesarios para producir el sistema deseado.

¹⁸ La ruta crítica es aquel camino en el cual un retraso en cualquiera de sus actividades provocaría un retraso en todo el proyecto.

La precisión de las estimaciones depende de las habilidades, conocimientos y experiencia del personal que prepara las estimaciones. Además, depende de la productividad del recurso humano que trabajará en el proyecto, la complejidad del sistema, las características y disponibilidad del usuario, las interrupciones ocasionadas por entes externos al proyecto, etc.

La estimación de costos en el desarrollo de sistemas, la medición de la productividad de los programadores y la medición de la calidad del software son unas de las áreas en las cuales se está realizando más investigación actualmente. Esto es esperanzador, pues las técnicas actuales tienen cierto parecido con la alquimia.

Hay dos razones que resumen la dificultad en la estimación:

- falta de identificar todas las tareas (un pecado de omisión). Esto ayuda a solucionarlo la EDT.
- estimación inexacta de las tareas identificadas (un pecado de comisión). Esto lo se discute adelante.

Al inicio de cada fase se revisarán las estimaciones preliminares contenidas en el plan. Esta revisión se basará en la experiencia acumulada durante el proyecto, una revisión de la EDT, con una lista más detallada de las actividades por emprender y de los recursos necesarios para poder llevarlas a buen término.

Esta estimación en firme se hace en ese momento, pues entonces se conocerán los productos, de fases anteriores, que son requisito para iniciar la fase. Con esa información, y con la experiencia de otros proyectos, será posible hacer una lista exhaustiva de actividades de la fase.

Para algunas fases, en especial la de codificación y prueba de unidades (programación), existen modelos y fórmulas que permiten calcular - con una precisión aceptable - el esfuerzo requerido.

3.1 ¿Quién hace la Estimación? .c3.

En torno a la responsabilidad sobre la estimación las opiniones están muy divididas. Hay quienes abogan porque sea una responsabilidad exclusiva del director del proyecto.

Otros prefieren que sea una responsabilidad de todo el equipo del proyecto. Aun otros juzgan que debe formarse un grupo de especialistas en estimación ([De Marco82]).

Un enfoque sencillo puede ser, entonces, que el director del proyecto haga una estimación inicial de la duración de las tareas. Seguidamente, el director pedirá a cada miembro del equipo que estime cuánto durará en las tareas que le son asignadas. El director comparará estas estimaciones con las suyas y pedirá justificar cualquier diferencia. Luego, el miembro del equipo adquirirá un **compromiso formal** (por escrito) de completar la tarea en el plazo que estipule el director ¹⁹. El director resuelve las discrepancias, las hace del conocimiento del equipo del proyecto y completa el plan.

Es deseable que se dé una formación básica en técnicas de estimación a algunos miembros del departamento de desarrollo de sistemas, así como a los actuales (o potenciales) directores de proyectos. Su misión será lograr exactitud en la predicción de esfuerzos y sistematizar la información sobre la productividad en el desarrollo de sistemas de la empresa.

3.2 Algunas Técnicas de Estimación .c3.

¹⁹ Para darle mayor formalidad al compromiso, puede pensarse que la empresa imponga sanciones a los empleados que incumplan con sus compromisos en un proyecto. La severidad de la sanción dependerá de la gravedad del incumplimiento, pero podría llegar hasta el despido sin responsabilidad patronal.

Hay varias técnicas de estimación del esfuerzo requerido para desarrollar sistemas. Ninguna de ellas ha comprobado ser superior a las otras (ver sección 3.3).

La técnica más común es la del **juicio experto** (o intuitiva). Esta técnica se basa en la experiencia y conocimiento previo del área por desarrollar que tengan uno o varios individuos de la empresa. Se procede de arriba hacia abajo, identificando los costos a nivel del sistema, luego los costos asociados al control de calidad, integración, documentación y entrenamiento. Normalmente no se utiliza documentación histórica de proyectos anteriores.

El empleo de la **técnica Delfi** (Delphi) permite que varios expertos, con base en la definición del sistema, hagan una estimación de los costos. Los expertos no interactúan, sino por intermedio de un coordinador. Luego, cada estimador conoce los resultados del promedio de todas las estimaciones y reconsidera su estimación. Se busca que los criterios converjan; cuando una estimación difiere mucho del promedio, se solicita una justificación de la misma.

El resultado de las dos técnicas anteriores se expresa comúnmente en **meses-persona** ²⁰.

Las **técnicas algorítmicas** parten de una medida cuantitativa, obtenida usando alguna de las técnicas anteriores, para predecir el número de meses-persona (MP) del proyecto, el tiempo total de desarrollo y el número de técnicos necesarios (en función de cada fase del ciclo). Se complementan con información "estándar" de la industria del software, que se ha ido recopilando desde los años setenta ²¹.

²⁰ Un mes-persona equivale al esfuerzo realizado por una persona durante un mes. En labores de programación, normalmente equivale a 152 horas (por mes). Deben tomarse en cuenta feriados, ausencias por enfermedad, permisos, etc.

²¹ Lamentablemente, en nuestro medio no se ha recopilado información sobre productividad en el

Casi todas las técnicas algorítmicas (como COCOMO o SLIM) parten de una estimación del número de líneas de programa que tendrá el producto final, lo cual ha sido ampliamente criticado (ver [Jones86]).

Como alternativa a las líneas de programa, pueden usarse los "puntos de función" ²², que toman en cuenta factores como los siguientes:

- nivel de experiencia del programador;
- nivel de complejidad del programa;
- grado en que el programador comprende el problema (programa) específico.

Una vez obtenidos los puntos de función, se usan en forma análoga a las líneas de código, como medida de productividad y como dato de entrada para la estimación. Existen algunas tablas que correlacionan el número de enunciados (instrucciones) de un lenguaje de programación que serían necesarios - en promedio - para codificar un punto de función ([Jones86]).

3.3 Problemas y sus Posibles Soluciones .c3.

Uno de los mayores problemas en la estimación es que no se ha estudiado mucho la productividad de las fases de especificación (análisis) y de diseño (arquitectónico y detallado). Esto se complica por la carencia de unidades estándar para medir la productividad en el desarrollo de software.

desarrollo de sistemas. La información "estándar" ha sido recolectada en Europa, Estados Unidos y Japón, principalmente.

²² Estos pretenden medir la "funcionalidad" o "utilidad" de un programa o sistema. Los puntos de función se relacionan empíricamente con el tipo y la complejidad del sistema de información por desarrollar.

La clave para el éxito en la estimación es tener un conocimiento de la productividad de la empresa en cuanto al desarrollo de sistemas. Para ello hay que recopilar datos y hay que saber qué se debe recopilar. Debe conocerse con claridad los detalles técnicos y políticos que influyen sobre la dedicación de esfuerzos al proyecto, la disponibilidad de recursos y el empleo del tiempo.

Todo proyecto debería destinar una fracción del esfuerzo a la recopilación de información clave para determinar la productividad y las fuentes de problemas en el desarrollo. Además de la documentación técnica, el proyecto deberá llevar una bitácora del desarrollo y producir un **legado del proyecto**, que permita su evaluación y sirva como referencia para esfuerzos futuros.

Cuando se hace la estimación, hay que tener en cuenta que pueden obtenerse **ahorros** en el desarrollo de sistemas mediante:

- el concurso de **personal bien calificado** ²³,
- el uso de **estándares** de desarrollo,
- la **reutilización de componentes** ²⁴,
- la aplicación de herramientas para **automatizar** algunas fases del **desarrollo**, y
- el empleo de **estrategias opcionales** para desarrollar los sistemas (prototipos, comprar el sistema, contratar implementación, etc.; ver capítulo 6).

4 Presupuesto .c2.

²³ La variabilidad es tanta que se han encontrado diferencias de productividad de 25 a 1 entre los programadores de una misma empresa ([Brooks75]). Son más típicas, sin embargo, relaciones de 4 a 1 en la productividad de los programadores de una misma empresa.

²⁴ En este caso, se busca volver a utilizar código existente, sea desarrollado por nosotros, o adquirido. Puede representar ahorros de hasta el 20% en el tiempo de desarrollo [Fairley85]. El uso de "plantillas" o "machotes" también puede proveer ahorros significativos.

Los costos se relacionan con los recursos aplicados directamente al desarrollo del sistema en sí. Sin embargo es frecuente descuidar otros aspectos, como:

- captura de los datos iniciales,
- entrega e instalación de equipos,
- operación del sistema nuevo,
- mantenimiento del sistema,
- operación del sistema existente (si existe uno que se vaya a desplazar).

Para cada uno de esos aspectos (directos o indirectos), hay que detallar los costos relacionados con los siguientes rubros:

- recursos humanos: salarios, cargas sociales, horas extras, etc. Se debe tomar en cuenta los recursos técnicos, los del usuario y los contratados externamente;
- hardware de cómputo y de comunicaciones (compra, alquiler);
- software: compra o alquiler de software de base, herramientas de desarrollo, software de aplicación, etc.;
- suministros: discos, cintas, papel, etc.;
- otros ítemes.

Cuando es previsible que el proyecto se prolongue por varios años, deben incluirse ajustes a los costos, al tomar en cuenta la inflación ²⁵.

Dado que la actividad de desarrollo de sistemas es intensiva en el uso de recurso humano, éste es el principal factor de costo en la construcción de sistemas.

Sin embargo, al desarrollar proyectos, los siguientes elementos no deben descuidarse, pues todos ellos consumen tiempo:

- las revisiones;
- la documentación ²⁶;

²⁵ Estos proyectos son peligrosos. Si el proyecto debe ser largo, sepáreselo en etapas.

²⁶ Lamentablemente casi siempre se descuida o se deja para el final. Debe ser realizada por el director, los

- la validación;
- la planificación;
- el entrenamiento de usuarios;
- la especificación de nuevos equipos;
- la instalación del sistema;
- la instalación de equipo;
- la conversión de bases de datos (o archivos);
- la capacitación del personal de informática en nuevos métodos, herramientas o lenguajes;
- la evaluación post-proyecto.

5 Otras Actividades que Requieren Planificación .c2.

Hasta el momento, la discusión ha girado en torno a la construcción de un sistema de información. Ahora se mencionarán otras actividades que deben planificarse, pues necesitan recursos (véanse [Fairley85], [NCC87] o [Pressman87] para obtener información detallada).

La **administración de la configuración** se refiere al control de cambios en los productos finales y en la documentación del sistema. Busca mantener versiones actualizadas en una biblioteca, así como el registro de su situación. Muchas "herramientas CASE" ²⁷ proveen facilidades para administrar configuraciones de software.

El **control de calidad** desarrolla los estándares para el proyecto y supervisa que se respeten, lleva a cabo inspecciones de procesos y productos, y realiza las pruebas de aceptación, en coordinación con el usuario.

programadores, diseñadores y analistas en todas las fases del ciclo de desarrollo del sistema.

²⁷ CASE (del inglés "computer-aided software engineering) es un nombre genérico para el soporte automatizado a distintas fases del desarrollo de sistemas. Las herramientas CASE permiten eliminar mucho del tedioso trabajo manual que conllevan varios métodos y técnicas para desarrollar software.

Las **herramientas y técnicas específicas** 46
por emplear en cada fase deben ser anticipadas en la etapa de planificación, pues requieren tiempo para su aprendizaje y uso.

6 Contingencia .c2.

Ningún plan estará completo si no se da una consideración detallada a los riesgos posibles, así como a las maneras de minimizarlos, mediante planes de contingencia, durante la implementación del proyecto.

El riesgo se minimiza mediante una planificación detallada de las actividades del proyecto, en la cual se garantice su completa comprensión y se examinen las consecuencias de su retraso o fracaso. También deben tomarse en cuenta todos aquellos entes (unidades organizacionales, personas, otras empresas) que puedan influir al proyecto.

6.1 El Plan de Contingencia .c3.

El propósito de un plan de contingencia es que el proyecto siempre sea capaz de producir un sistema de información viable, a pesar de los contratiempos que pueda encontrar.

El plan de contingencia contendrá acciones concretas que deben emprenderse a fin de reducir el riesgo o cuando el proyecto se desvíe de lo planeado.

Deben considerarse dos tipos de contingencias: **problemas identificables** que pueden surgir y **problemas imprevistos**.

Para los problemas identificables debe indicarse las acciones por emprender, así como cuánto tiempo y costo agregarán al proyecto. Por ejemplo, si la computadora necesaria para las pruebas no está disponible en la fecha especificada por el plan, habrá que definir dónde se harán las pruebas.

Los imprevistos son difíciles de atacar, porque no se puede dar respuestas a problemas indefinidos. Una posible manera de reducir el número de imprevistos es ser realista (y no idealista) en la planificación de recursos. El personal puede enfermarse, renunciar, cometer errores, no comprender una especificación; las computadoras fallan; puede no lograrse el entendimiento con el usuario...

Es ideal que el director pueda recurrir a los archivos de proyectos pasados e investigar sus legados. En ellos encontrará la historia de los problemas enfrentados por la empresa en el desarrollo de sistemas anteriores, así como las soluciones encontradas. Muchos de esos problemas bien pueden ocurrir de nuevo, por lo que podrá prevenirseles. Esta es otra de las razones por las cuales es conveniente que, en nuestro medio, los proyectos produzcan un **legado** y sean sometidos a una evaluación *a posteriori*.

A partir de su propia experiencia y de otra información disponible, el departamento de informática puede ir desarrollando listas de comprobación, lo más exhaustivas posible, que detallen la descomposición de las actividades y la atención de los problemas posibles durante el desarrollo de un sistema de información. Esto facilitará significativamente la planificación de los proyectos subsiguientes.

Es conveniente elaborar una **matriz de riesgos** para cada proyecto. Abajo se muestra un ejemplo, que incluye algunas áreas de riesgo y su posible mitigación ([Shere88]):

6.2 Identificación y Reducción del Riesgo .c3.

Area crítica	Síntoma	Mitigación de riesgo
Requerimientos cambiantes	No se realizan funciones necesarias. Excesos en costo y tiempo.	Participación efectiva del usuario. Validar requerimientos en cuanto a necesidades y factibilidad.
Falta de documentación	"Parches" causan fallas. Mantenimiento costoso. Pronunciada curva de aprendizaje.	Documentar durante desarrollo y mantenimiento. Biblioteca centralizada para documentación CASE. Auditorías e inspecciones de documentación.
Requerimientos no definitivos	Mala interpretación conduce a implementar funciones equivocadas. Presupuesto no justificado. Mala relación con el usuario.	Especificar requerimientos medibles antes del diseño. Usar prototipos rápidos. Educar al usuario.

La evaluación del impacto de las áreas críticas debe ser tan específica como sea posible. De ser factible, el riesgo debe expresarse en términos de las pérdidas (en dinero) que provocaría.

Las matrices, como la mostrada anteriormente, son muy útiles durante la planificación y las revisiones de avance. Cuando se planea el proyecto, conviene colocar - en el sitio de reuniones - una gran matriz en blanco, al alcance del director del proyecto y del personal más experimentado del equipo. Cada vez que alguno de ellos piense en una área de riesgo, la escribe - espontáneamente - en la matriz.

El equipo del proyecto y los usuarios luego evaluarán los riesgos y buscarán maneras de mitigarlos. También identificarán soluciones de respaldo, en caso de que, a pesar de reducir el riesgo, las cosas salgan mal. Para estas sesiones pueden usarse diversas técnicas de grupo, entre otras: "tormentas de ideas", técnica del grupo nominal ²⁸, sinéctica ²⁹ y descomposición de problemas.

Entre las técnicas para reducir el riesgo, se pueden mencionar:

²⁸ Esta técnica es similar a la Delfi. Se la usa para dilucidar opiniones y agregar apreciaciones. Permite identificar metas y establecer prioridades, involucrando al personal en el proceso de toma de decisiones (incrementando la posibilidad de aceptación final del producto).

²⁹ En ésta, el grupo identifica posibles soluciones a los problemas, al adaptar soluciones de otras disciplinas o de experiencias pasadas

- asegurar la participación del usuario y el respaldo de la administración superior;
- construir prototipos;
- ofrecer entrenamiento y educación (a usuarios y a técnicos);
- generar documentación fácilmente mantenible;
- hacer partícipes, en la evaluación y prevención de riesgos, al usuario y al equipo del proyecto;
- usar mejores procedimientos de verificación y validación;
- utilizar herramientas para el desarrollo de sistemas, que reduzcan los errores y el trabajo manual (CASE, generadores de aplicaciones);
- involucrar a la auditoría desde el desarrollo del proyecto (y no *a posteriori*).

7 Presentación del Plan .c2.

Previa a la planificación propiamente, se hace una selección del sistema por desarrollar (aquí llamada "evaluación inicial", ver capítulo 1). Es conveniente que, a partir del plan maestro de sistemas de información ³⁰, se elabore un breve documento, la **definición del sistema**, que contenga ³¹:

- Definición del problema;
- Justificación del sistema;
- Metas del sistema y del proyecto;
- Restricciones del sistema y del proyecto;
- Funciones que se proveerán;
- Características del usuario;
- Ambientes en que se desarrollará, operará y mantendrá el sistema;
- Estrategias consideradas para la solución;
- Prioridades para las diversas características del sistema;
- Criterios para la aceptación del sistema.

El **plan del proyecto**, propiamente, puede seguir el siguiente formato:

- Modelo del ciclo de vida que se seguirá. Terminología, hitos, productos por fase y finales;
- Organización del equipo y su administración;
- Estructura del desglose del trabajo;
- Necesidades de personal y otros recursos;
- Calendario preliminar;
- Estimación preliminar de costos;
- Mecanismos de supervisión y control del proyecto;
- Herramientas, técnicas, estándares, lenguajes por usar;
- Plan de pruebas;
- Documentación que producirá el proyecto;
- Formas de demostración y entrega de productos;
- Calendario y materiales de entrenamiento;
- Plan de instalación y entrega final;
- Consideraciones de mantenimiento.

Como se ve, el formato del plan del proyecto es extenso. Esto se justifica para la mayoría de los proyectos, pero pueden usarse variantes simplificadas para documentar el plan de un proyecto pequeño .

Si se cuenta con una disciplina para la administración de proyectos, estándares para documentación, legados de proyecto y estándares técnicos, se hará más fácil la concepción y documentación del plan.

En [Metzger81] se expone de manera detallada los diversos aspectos y contenidos de un plan de proyecto. En el capítulo 2 de [Fairley85] se muestran formatos para otros documentos de un proyecto de sistemas de información.

³⁰ Ver [Bondu89]

³¹ Éste y el formato siguiente (plan del proyecto) han sido adaptados de [Fairley85]

VI Estrategias para el Desarrollo de Sistemas .c1.

En los capítulos precedentes se ha descrito la planificación de los sistemas de información desarrollados mediante los métodos tradicionales. Sin embargo, existen numerosas opciones para obtener sistemas de información, que serán estudiadas en este capítulo. El poder usarlas adecuadamente depende de factores como:

- las características intrínsecas de la aplicación por desarrollar;
- la madurez de la empresa en el uso de sistemas de información y de la tecnología informática;
- la disponibilidad de recursos para adquirir productos (sistemas terminados, contratación, herramientas);
- la pericia del personal de informática;
- el desarrollo propio de la tecnología informática, su disponibilidad, conocimiento y soporte en nuestro medio;
- la disponibilidad de productos que satisfagan las necesidades de las aplicaciones por desarrollar;
- la naturaleza de los competidores y del entorno.

Luego de hacer algunas consideraciones generales, se examinará someramente las principales opciones para desarrollar (u obtener) sistemas de información ³². En este capítulo se dará énfasis en los aspectos relacionados con el **proceso** de desarrollo, sin dar consideración a las particularidades del **producto** que se espera obtener ³³.

³² Se hace caso omiso de las metodologías enfocadas al desarrollo de sistemas de información comunes. Esto se describió ampliamente en los capítulos anteriores.

³³ El producto puede ser un sistema de información administrativo común, un sistema de apoyo a las decisiones, un sistema basado en conocimientos (ver [Gutiérrez89]), una tecnología para automatización de oficinas, etc.

1 Selección de una Estrategia para Desarrollar el Sistema .c2.

En [Gremillion83] se sugieren tres propiedades importantes de un sistema de información, que deben considerarse en la selección de una estrategia adecuada para su desarrollo. Estas son tratadas seguidamente.

Los **rasgos comunes** o "**regularidad**" ³⁴, que se refieren al grado con el que otras empresas podrían utilizar la misma solución para un determinado problema. Si el problema es común a muchas empresas, existe una alta probabilidad de que el sistema deseado ya esté disponible en el mercado. Las preguntas clave que deben hacerse son:

- ¿Tienen la mayoría de nuestros competidores esencialmente el mismo problema que nosotros?
- Si nuestra solución es diferente a la de nuestros competidores, ¿Por qué lo es?
- ¿Es reducido el costo de adaptar nuestra empresa a la solución común?

Segundo, el **impacto**, que es el grado hasta el cual el sistema afectará la empresa. Éste determinará el grado de participación que deberían tener los profesionales de informática en el desarrollo del sistema. A mayor impacto, habrá mayor necesidad de participación profesional para asegurar la calidad del sistema. Deberá determinarse:

- ¿Es amplia la utilidad del sistema, o se confina a unas pocas personas?

³⁴ Así se decidió traducir el término "commonality". La palabra "comunidad" es una traducción correcta pero, por poseer otras acepciones en español, no se ha usado. Otras traducciones posibles son: "frecuencia" y "ordinariedad".

- ¿Responde el sistema a problemas estratégicamente importantes?
- ¿Qué impacto tendría en la empresa una falla del sistema?

Por último, la **estructura**, que se refiere a cuán bien se comprenden el problema y su posible solución. A menor certidumbre acerca de lo que el sistema debe hacer, se hará más necesaria la participación del usuario en el proceso de desarrollo. Las preguntas pueden ser:

- ¿Cuán bien definido se encuentra el proceso que este sistema apoyará?
- ¿Cuán confiablemente pueden especificarse con anticipación las entradas y salidas necesarias para el sistema?

El siguiente esquema puede orientar la toma de decisiones sobre la estrategia de desarrollo por seguir:

Propiedades del proyecto			Método sugerido
Regularidad	Impacto	Estructura	
Común	Amplio	Alta	Paquete
No común		Amplio	Tradicional
No común		Amplio	Prototipo
Común	Limitado	Alta	Paquete
No común		Limitado	Usuario final
No común		Limitado	Usuario final

La administración superior (el comité ejecutivo de automatización) tiene un papel importante que jugar en cuanto al incremento de la productividad en el desarrollo de los sistemas de información, pues puede definir políticas que fomenten el establecimiento y puesta en marcha de nuevas estrategias para los sistemas de aplicación informáticos.

En este respecto, una clara dirección por parte de la administración superior le dará al departamento de informática la confianza necesaria para utilizar estrategias alternas al modelo tradicional del desarrollo de sistemas.

El lector advertirá que las estrategias presentadas no tienen por qué ser excluyentes entre sí. Por ejemplo, se puede adquirir un paquete y usarlo como punto de partida para desarrollar un sistema que tendrá mayor funcionalidad, que luego la empresa utilizará para ofrecer nuevos servicios. Asimismo, se puede partir de un sistema desarrollado por el usuario final, como primer prototipo de un sistema que será refinado sucesivamente por un equipo de proyecto.

A continuación se presentan algunos detalles sobre las estrategias opcionales para desarrollar sistemas, la relación que guardan con el modelo tradicional (de "cascada") y la manera en que se afectan la planificación y el control del proyecto.

2 Adquisición de Sistemas Pre-Hechos (Paquetes) .c2.

Los "paquetes" son soluciones a problemas que están parcial o totalmente listas para ser implementadas. El propósito de un paquete es reducir parte del esfuerzo dedicado al diseño y la totalidad del destinado a la programación. Esto no es posible siempre, pues algunas organizaciones desean modificar el paquete, con el fin de ajustarlo a su ambiente y necesidades peculiares.

Las **ventajas** de los paquetes incluyen:

- duración total del proyecto menor que la del desarrollo propio;
- costos totales inferiores, pues el paquete se vende a varias organizaciones, entre las cuales se distribuye el costo de desarrollo;
- pueden tener mayor funcionalidad que los sistemas propios;
- pueden tener menos errores, si el paquete ya ha sido vendido y se encuentra operando en otras empresas.

Entre sus **desventajas**, se encuentran:

- el paquete puede no tener todas las funciones requeridas por el comprador;
- puede existir la necesidad de cambiar extensivamente el paquete. Si el vendedor se rehúsa a hacerlo, el personal de la empresa compradora puede tener dificultad en comprender y cambiar el paquete;
- el comprador puede verse obligado a cambiar sus procedimientos internos;
- el comprador depende del soporte que le dé el vendedor del paquete.

Aprovechando la regularidad de la aplicación, normalmente el vendedor construye un sistema generalizado, que satisfará las necesidades de muchos clientes. Es usual que el paquete incluya parámetros que le permitan adaptarse al ambiente del cliente. Por ejemplo, un paquete de inventarios puede incluir módulos para indicar la manera de manejar el costo del inventario (promedio, primero-en-entrar-primero-en-salir, último-en-entrar-primero-en-salir, etc.).

El ciclo de desarrollo se ve afectado en las fases de diseño (arquitectónico y detallado), programación (codificación y prueba de unidades) e integración, las cuales prácticamente desaparecen. El éxito en la adquisición del paquete depende, primordialmente, del **estudio de necesidades** que se haya hecho, previamente, al interior de la empresa.

Las actividades de la fase de especificación de requerimientos deberán asegurar que se incluya:

- una especificación precisa de las necesidades de información;
- jerarquizar las necesidades como *imprescindible* (la característica es obligatoria), *esencial* (la característica debería incluirse, a menos que sea costoso o difícil implementarla) o *deseable* (la característica puede ser beneficiosa, pero no degradaría significativamente al paquete);
- establecimiento de los criterios de aceptación del paquete;
- establecimiento de una propuesta de contrato con el eventual proveedor.

Después de hacer los contactos con los proveedores e invitarlos a cotizar (por licitación, contratación directa, etc.), se hará una **selección previa** para eliminar aquellos proveedores cuya descalificación sea obvia. Si ningún proveedor satisface los requerimientos, entonces se procedería a redefinir las necesidades o reconsiderar la decisión de comprar vrs. hacer.

La **evaluación final** será más concienzuda, e incluiría:

- **evaluación de usuarios**, se visita a clientes existentes del paquete, evaluando su satisfacción;
- **evaluación técnica**, que examina los siguientes aspectos del paquete: satisfacción de las necesidades, rendimiento aceptable, documentación, mantenibilidad, seguridad, formatos de archivos, etc.;

- **evaluación comercial**, que evalúa al proveedor como empresa (credibilidad, permanencia, solidez, calidad del soporte, perspectivas futuras).

Se señalan a continuación algunas recomendaciones adicionales. El lector interesado puede consultar [Gremillion83], [Gilhooley85], [Martin83] y [Lucas87], donde encontrará información más detallada sobre este respecto.

Es importante tener en cuenta si en la empresa existe una **base de datos** (o si está planeada para el futuro cercano). De ser así, habrá que determinar si la nueva aplicación tendrá relación cercana con otros sistemas. El administrador de la base de datos deberá indicar si es posible establecer un "puente" entre el paquete y la base de datos.

El **mantenimiento** puede resultar difícil. En algunos casos, porque no se provee el código fuente de los programas, en otros, porque la empresa hace cambios en sus sistemas administrativos (o se ve sometida a regulaciones gubernamentales) que hacen imposible que el paquete se ajuste a las nuevas condiciones. Si es previsible que esto suceda para el área de aplicación, es más razonable pensar en desarrollar internamente el sistema. En cualquier caso, si el paquete incluye generadores de informes o lenguajes de consulta, la flexibilidad es mayor y el riesgo de obsolescencia se reduce.

De particular importancia es la firma de un **contrato** entre el comprador y el proveedor. Muchos proveedores poseen machotes de contratos. Estos contratos deben ser revisados con cuidado por el comprador, para adaptarlos de manera que resguarden sus intereses.

Es posible que el proveedor quiebre, cambie la orientación de su negocio, o decida suspender el soporte al paquete adquirido. Por ello, es conveniente incluir en el contrato una cláusula que proteja a los clientes ante esa eventualidad. El cliente debe insistir en que el proveedor mantenga en **resguardo** una copia del código fuente, la documentación técnica y las herramientas de software relacionados con el paquete. El lugar donde se guarde esto lo establecerían de común acuerdo el comprador y el proveedor.

Sin embargo, siempre es preferible que el proveedor entregue al comprador una copia completa del paquete (código fuente y su documentación asociada). El comprador debe insistir en hacer efectiva esta opción.

Por último, es conveniente que el comprador **ensaye** con cuidado los paquetes que le son ofrecidos por los proveedores. Aún cuando ya se ha decidido a quién comprarle, es importante que se establezca un período de prueba, de manera que el comprador se asegure que el paquete responde a sus necesidades adecuadamente, con derecho a devolverlo y lograr un reembolso de su inversión.

3 Sistemas Contratados .c2.

Los sistemas contratados o "a la medida" son semejantes a los paquetes, en el sentido de que se recurre a entidades externas para que resuelvan total o parcialmente la producción de un sistema de información.

En este caso, la empresa tiene una diversidad de opciones, que varían según las tareas que desee desarrollar por sí misma. Es posible que se contrate a un proveedor para que desarrolle un sistema de información, partiendo desde la fase de especificación de requerimientos (análisis) o después de ella. También es posible la contratación externa de la programación del sistema, con base en el diseño detallado.

Los principales **peligros** que se corren son similares a los del ciclo de vida, es decir, que el proveedor se retrase en la entrega del producto o que su calidad (satisfacción de necesidades, eficiencia, documentación) sea pobre. El factor de costo normalmente no es un problema, pues se establece como una condición de contratación.

Por estas razones, es recomendable que el comprador contrate un sistema a partir de necesidades claramente establecidas. Además de éstas, puede añadir lo indicado en la sección 6.2 (otras actividades de la fase de especificación de requerimientos).

Cuando el comprador tiene recursos humanos insuficientes para realizar la especificación, puede recurrir a un consultor externo. El consultor externo deberá contar con un apoyo decidido de la administración superior y del eventual usuario del sistema, de otra manera su trabajo será inservible.

La **evaluación** del proveedor (es decir, la empresa que desarrollará el sistema) se hará con base en los siguientes parámetros:

- experiencia general en el desarrollo de aplicaciones;
- experiencia en aplicaciones similares;
- reputación de la empresa (seriedad, puntualidad, calidad, solidez, permanencia);
- comprensión de los requerimientos específicos de la aplicación por desarrollar;
- características de la propuesta que presenta (costo, facilidades, documentación, garantías, entrenamiento a usuarios, mantenimiento).

La **aceptación** de sistemas desarrollados bajo esta modalidad sólo se hará después de un riguroso procedimiento de pruebas, que incluya una "en vivo" durante un plazo razonable (se recomienda un mínimo de 30 días) y hasta después de haber entrenado a los usuarios. Los pagos por el sistema no deberán completarse sino hasta que el producto sea finalmente aceptado.

Al tiempo que se elabora el **contrato de desarrollo** deberá hacerse un borrador del **contrato de mantenimiento** o, en su defecto, especificar quién es el propietario del código fuente y de la documentación del sistema. Si el proveedor no entrega estos últimos, procédase de manera similar a la indicada en la sección 6.2.

Es deseable incluir, como parte del contrato, **castigos** al proveedor en el caso de incumplimiento en los plazos. Asimismo, se pueden anotar premios por entregas tempranas, siempre que se satisfagan los requerimientos del comprador.

De manera similar a lo indicado en capítulos anteriores, la experiencia ha mostrado que hay mayores garantías de éxito cuando la empresa compradora asigna un equipo de trabajo para que colabore con el proveedor en la especificación del producto. Este mismo equipo podrá atender las fases de pruebas del sistema, previas a su aceptación final. Cuando el proveedor emplea alguna metodología o técnica novedosa, el personal asignado por la empresa compradora quedará debidamente capacitado para aplicarlas al mantenimiento del sistema (y al desarrollo de nuevos sistemas en el futuro).

[Price86] hace una presentación detallada de este tipo de proyectos, desde los puntos de vista del comprador y del proveedor, acompañados por listas de comprobación, procedimientos administrativos y cláusulas contractuales para cada parte. [Lustman84] expone brevemente algunos aspectos relacionados.

4 Prototipos y Generadores de Aplicaciones .c2.

En la mayoría de las ingenierías, los prototipos y modelos permiten reducir el riesgo de producir un objeto que:

- no satisfaga las necesidades del usuario;
- es demasiado costoso;
- no tiene el rendimiento requerido;
- es poco confiable.

Los riesgos pueden ser de aplicación o técnicos. Los riesgos de **aplicación** comprenden productos que no satisfacen las necesidades del usuario, sea porque el usuario no tiene suficiente experiencia en sistemas informáticos, ora porque los desarrolladores no conocen suficientemente el área de aplicación.

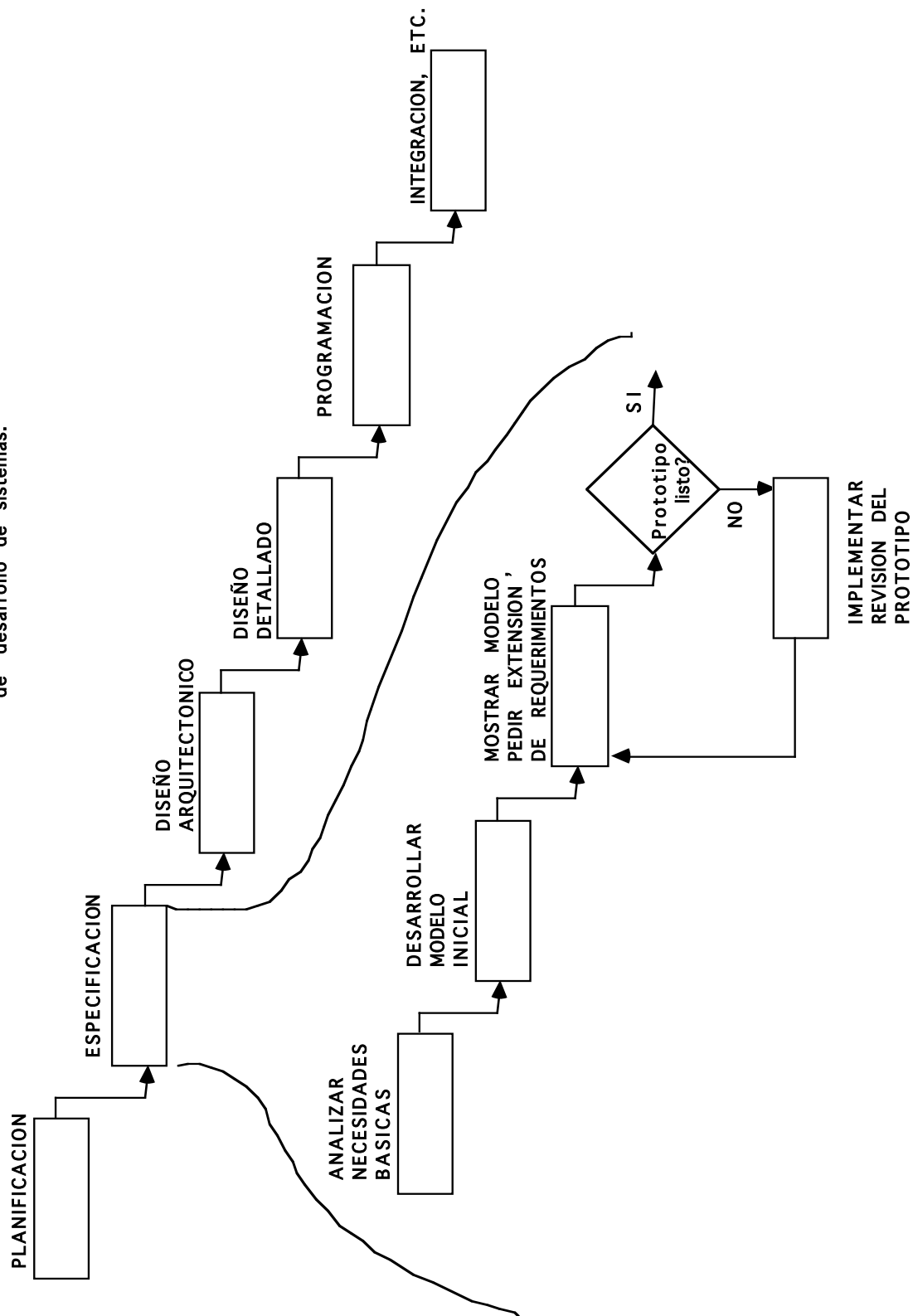
El riesgo **técnico** surge cuando la complejidad o novedad del sistema le hace difícil de desarrollar.

Hay una creciente evidencia de que ambos tipos de riesgo pueden reducirse mediante la construcción de prototipos de sistemas.

Para que el prototipo sea eficaz, el costo de producirlo deberá ser únicamente una parte del costo de producir un sistema con las características de calidad completas. Esto puede lograrse reduciendo la funcionalidad del sistema modelado, o bien usando **generadores de aplicaciones**, que permiten construir rápidamente sistemas de funcionalidad limitada (que a veces exhiben un pobre desempeño).

Los prototipos **exploratorios** modelan, total o parcialmente, las funciones e interfaz de usuario del sistema por construir. Este modelo se emplea, con la participación del usuario, para aclarar iterativamente los requerimientos que el sistema deberá satisfacer. Estas aproximaciones sucesivas permiten reducir el riesgo de aplicación y modifican el ciclo de desarrollo únicamente en lo concerniente a la especificación de requerimientos, como se aprecia en la figura 9.

Figura 9 El uso de prototipos en el ciclo de desarrollo de sistemas.



Los prototipos **experimentales** se utilizan en la fase de diseño, con el propósito de evaluar distintos algoritmos o enfoques, de los cuales no se conocen algunas propiedades. Los prototipos experimentales reducen el riesgo técnico.

Los generadores de aplicaciones, "**lenguajes de cuarta generación**" y otros tipos de herramientas (escritores de informes, diseñadores de diálogos, verificadores de entrada de datos, esqueletos de programas) ayudan a construir rápidamente módulos de software que cumplan con la funcionalidad especificada en los requerimientos.

Su limitación fundamental es que pueden resultar rígidos para ciertos requerimientos que no encajan exactamente en su "modelo de procesamiento". A menudo hay que readecuar el diseño de un sistema para ajustarlo a las particularidades de un lenguaje de cuarta generación.

Los prototipos, pues, constituyen una manera de establecer los requerimientos del sistema. De ninguna manera debe verse a los prototipos como sustitutos de los métodos analíticos que se emplean para dilucidar los requerimientos. Más bien pueden ser complementarios, al permitir que se **valide** el trabajo analítico mediante el prototipo.

Otra técnica que parece promisoriosa es la de los **lenguajes de especificación ejecutables** (o requerimientos ejecutables). Estos permitirían ejecutar los requerimientos y simular el comportamiento del sistema en etapas tempranas del proyecto, pudiendo así reducir, o hasta evitar, las etapas de pruebas en el proyecto.

El factor clave del éxito en el uso de prototipos es la **involucración del usuario**, quien puede apreciar rápidamente algo concreto como resultado del esfuerzo de desarrollo. Con el enfoque tradicional es difícil lograr que los usuarios entiendan realmente las especificaciones del sistema, pues la primera vez que interactúan con él es durante las fases de prueba.

Con un prototipo, el analista-diseñador obtiene retroalimentación rápida de parte del usuario, aproximándose así más eficazmente a definir los requerimientos del sistema. Como los usuarios examinan el prototipo, ellos participan efectivamente en el proceso de desarrollo.

[Alavi84] da las siguientes guías para la aplicación del enfoque de desarrollo basado en prototipos:

- **Quién:** los diseñadores y usuarios deben conocer la filosofía del enfoque de prototipos (de otra manera, se generará mucha frustración);
- **Dónde:** en ambientes donde se cuenta con respaldo para el enfoque de prototipos, con diseñadores y usuarios motivados, y con herramientas técnicas adecuadas;
- **Cuándo:** cuando los requerimientos del usuario son difusos o ambiguos;
- **Cómo:** se requiere una planificación rigurosa del costo y del tiempo, así como el establecimiento y cumplimiento de procedimientos para el control y la coordinación.

Cabe advertir que el director del proyecto debe hacer comprender a los usuarios que el prototipo es un **medio** para conseguir un sistema más adecuado para ellos, pero que no es el sistema final. Los usuarios pueden generar muchas expectativas y tornarse impacientes, pues ya han visto objetos que se comportan como un producto terminado, pero que no son más que un cascarón, es decir, la interfaz con el usuario.

Otra actitud que se ha observado es que el usuario pide cambios muy frecuentes en el prototipo, causando desazón en el analista que trabaja con él. La solución para ello es aplicar con mayor conciencia los métodos del análisis de sistemas y lograr un compromiso, por parte del usuario, con la versión definitiva de la interfaz.

En [Alavi84], [Alavi85], [Boar86], [Gilhooley86], [Lee86], [Lucas87], [Pressman87], [Lansman88], [Campbell87] y [Lee88] se tocan en detalle los temas de prototipos, generadores de aplicaciones y lenguajes de cuarta generación. [Boar84] trata extensivamente el tema de los prototipos, su papel en el desarrollo de sistemas y su impacto en la organización.

5 Desarrollo Incremental y por Fases .c2.

Es posible usar estrategias **radicales** para desarrollar sistemas de información. Como una extensión del desarrollo basado en prototipos, en el modelo **incremental** se construyen versiones sucesivas del producto, las cuales se van refinando y mejorando, a la vez que se les añade mayor funcionalidad.

La ventaja de este enfoque es que se contará rápidamente con un producto útil, que brinda resultados y que puede ser experimentado por el usuario. Sus dos variantes son:

- una fase de análisis (especificación de requerimientos) seguida por un proceso iterativo de diseño e implementación de versiones sucesivas;
- una fase de análisis seguida por el diseño arquitectónico de las diversas versiones, las cuales luego son construidas una a una.

Aunque parece reminiscente de un estilo de desarrollo de sistemas indisciplinado, que condujo a tantos fracasos en el pasado, cabe esperar que, con una planificación y control formales, así como con el apoyo de herramientas de desarrollo, el enfoque incremental puede ser administrado adecuadamente.

En particular, el enfoque incremental parece apropiado para sistemas con baja estructura (alto riesgo de aplicación) y bajo nivel relativo de tecnología (bajo riesgo técnico), en especial aquellas áreas de mayor potencialidad para los lenguajes de cuarta generación.

Una evolución del ciclo tradicional, los prototipos y el enfoque incremental lo constituye el **diseño por fases**, propuesto en [Dennis87].

En el diseño por fases se realiza la especificación de requerimientos y el diseño arquitectónico de la aplicación, dividiéndola en sus principales subsistemas. Luego se entra en un proceso iterativo de selección e implementación de los subsistemas.

Cuando el subsistema se selecciona (sobre la base de alguna priorización o precedencia lógica), se hace rápidamente su diseño detallado, codificación y prueba. Luego se somete a evaluación del usuario, lo que puede provocar modificaciones. Cuando ya no hay más cambios, puede mejorarse el prototipo construido de esta manera (mejorando el desempeño, la seguridad, etc.)

Puede ocurrir que, en este proceder, se identifiquen ciertas omisiones del diseño arquitectónico, las cuales deberán corregirse, así como todos los subsistemas afectados, los cuales se reimplementan. Cuando se entrega un subsistema, a satisfacción del usuario, se procede a seleccionar el siguiente subsistema.

El diseño por fases permite atrapar temprano la mayoría de los errores de diseño, lo que reduce el costo total del proyecto. Este enfoque resulta aplicable a aquellos sistemas que pueden ser particionados en subsistemas relativamente independientes, pues cada uno de ellos se desarrolla como un prototipo.

Entre las ventajas del diseño por fases están:

- los problemas del diseño original, así como las mejoras al mismo, pueden ser realizados por el mismo equipo de desarrollo, y no por un equipo de mantenimiento que no está familiarizado con la aplicación;
- la satisfacción del usuario es mayor, pues experimenta con el sistema y lo aprueba;
- el sistema se introduce gradualmente, lo que facilita su aprendizaje.

Los proponentes de este enfoque lo han aplicado, con éxito, a sistemas que poseen complejidad e incertidumbre altas.

6 Desarrollo por el Usuario Final .c2.

El advenimiento de los microcomputadores poderosos de bajo costo, dotados de software dirigido al usuario final "no iniciado" en programación, está transformando las empresas de hoy.

Con los microcomputadores y su software asociado, ha sido posible dotar, a los diversos individuos de la organización, de un vasto poder computacional, accesible desde su puesto de trabajo.

El amplio mercado de los microcomputadores ha estimulado el florecimiento de una industria de software altamente competitiva, en la cual la facilidad de uso, la flexibilidad y la eficiencia de los productos son claves para su éxito.

La introducción de los microcomputadores no resulta tan fácil para la administración. Los microcomputadores tienen la ventaja de ser independientes de los computadores centrales, pero se les dificulta el acceso a los datos corporativos. El computador central tiene el beneficio de la integración, que entra en conflicto con la independencia de los microcomputadores.

Los departamentos de informática ³⁵ han gravitado, tradicionalmente, en torno a los computadores centrales, caracterizándose por responder lentamente - o nunca - a las necesidades de información muy particulares de algunos usuarios.

Por esta razón, en muchas empresas se crearon los **centros de información**, con el propósito de dar apoyo al usuario en cuanto al empleo adecuado de las herramientas computacionales a su disposición (véase [Monge88]).

Muchos usuarios tienen el nivel de entrenamiento suficiente para desarrollar sus propias aplicaciones mediante el uso de paquetes o, incluso, mediante la programación. Los usuarios finales generalmente desarrollan sus aplicaciones iterativamente, al ir mejorando la percepción de sus propias necesidades y conocer mejor sus herramientas de trabajo.

Las ventajas potenciales de la informática del usuario final incluyen:

- acceso oportuno y mejor a la información;
- mejora en la toma de decisiones;
- mejor control por parte del usuario;
- menores costos de desarrollo de sistemas.

Sin embargo, esto puede entrar en **conflicto** con algunos intereses generales de la empresa.

Por ejemplo, la **documentación**. En algunos casos, desarrollar un sistema para un usuario puede requerir unos pocos minutos (por ejemplo, en una hoja de cálculo), mientras que la documentación bien puede demandar un tiempo mayor. En otros casos, los enfoques clásicos de documentación simplemente no son aplicables.

La necesidad de documentar este tipo de sistemas depende del tamaño del equipo que desarrolla la aplicación y del plazo del proyecto. Cuando hay varios participantes la comunicación entre ellos se hace necesaria.

Otro punto de conflicto es el **rendimiento** del sistema, pues el usuario puede desarrollar sistemas que no se desempeñen adecuadamente o que degraden el rendimiento de otros sistemas.

³⁵ O "centros de cómputo", como se aún se les llama con frecuencia.

Algunos **otros riesgos** incluyen ([Brancheau87]):

- el desarrollo de sistemas no confiables debido a la falta de control de calidad;
- el desarrollo de sistemas incompatibles, que no siguen las interfaces estándar;
- la incomprensión o falta de información, que provoca decisiones erróneas;
- la baja utilización de equipo caro;
- el desarrollo de sistemas no prioritarios;
- el desperdicio de tiempo en aprender o desarrollar sistemas inadecuados.

El personal de informática (o del centro de información) puede ayudar a mejorar la calidad de esos sistemas, si provee los siguientes servicios ([Ryan86b]):

- entrenamiento en el empleo de herramientas dirigidas al usuario final;
- evaluación de las aplicaciones propuestas para propósito de desarrollo por el usuario final;
- evaluación y selección de herramientas;
- asistencia y participación en la implementación;
- asistencia en el diseño y prueba de las implementaciones;
- entrenamiento básico en metodologías de desarrollo de sistemas de información.

Cabe notar que los usuarios más experimentados estarán en capacidad de desarrollar **prototipos** de sus aplicaciones, y darán la base para que los técnicos en informática les tomen el relevo con el propósito de reducir el riesgo de la aplicación (por ejemplo, integrarla) o completar el desarrollo de la misma, a partir del prototipo elaborado por el usuario.

Al igual que en el desarrollo de sistemas por parte del personal de informática, el desarrollo de sistemas por el usuario garantizará su éxito en la medida en que responda a las políticas y planificación de la empresa.

En [Couger86] se propone maneras constructivas de estimular el desarrollo de sistemas por el usuario, pero asegurando que éstos son relevantes y compatibles con el resto de la estrategia informática de la empresa, por ejemplo:

- **hardware seleccionado**, el departamento de informática da apoyo sólo a cierto tipo de computadores;
- **compras centralizadas**, consecución de descuentos en las adquisiciones; estos ahorros se trasladan a los departamentos usuarios;
- **capacitación limitada**, únicamente en el software que satisface los estándares del departamento de informática;
- **distribución de software consistente**, un ente central distribuye todas las nuevas versiones del software;
- **desarrollo distribuido**, enseñar a los usuarios-desarrolladores buenos métodos de desarrollo, que aseguren el cumplimiento de los estándares, faciliten el mantenimiento y reduzcan su costo;
- **bonificaciones de software**, como correo electrónico, para quienes tengan microcomputadores compatibles.

Este tipo de controles se paga con creces, si se considera el beneficio de limitar la proliferación y las incompatibilidades, reducir los problemas de mantenimiento y promover un ambiente creativo enfocado a reducir los costos.

Los siguientes artículos versan sobre la materia aquí resumida: [Brancheau87], [Couger86], [Gremillion83], [Huff88], [Keen84], [Monge88], [Pliskin87], [Ryan86a] y [Ryan86b].

VII Apéndice .C1. Evaluación del Riesgo de una Aplicación

Adaptado de [McFarlan81] y [Hammond86].		Ambiente altamente estructurado (requiere pocos cambios de procedimiento)	1
Evaluación del riesgo por TAMAÑO	factor x peso riesgo	Ambiente razonablemente estructurado	2
1. Tiempo dedicado al desarrollo del sistema	5	Ambiente poco estructurado (requiere muchos cambios de procedimiento y entrenamiento)	3
Menos de 15 meses-persona	1		
De 15 a 24 meses-persona	2		
Más de 24 meses-persona	3		
2. Duración estimada del proyecto	4	2. Si se propone un sistema de reemplazo	5
12 meses o menos	1	¿Cuál porcentaje de las funciones existentes son reemplazadas?	
13 a 24 meses	2	Del 0% al 25%	3
Más de 24 meses	3	Del 25% al 50%	2
		Del 50% al 100%	1
3. Personal que hará el trabajo	4	3. ¿Tiene la organización necesidad de cambiar su estructura para satisfacer los requerimientos de nuevo sistema?	5
Personal propio	1	No	0
Personal propio en porciones significativas	2	Mínimamente	1
Personal externo en casi todas las labores	3	Un tanto	2
		Mucho	3
4. Número de departamentos involucrados en el proyecto (sin contar informática)	4	4. ¿Cuál es la actitud general del usuario?	5
Uno	1	Antagónica a la solución automatizada	3
Dos	2	Un tanto renuente	2
Tres o más	3	Favorable, entiende las ventajas de la solución automatizada	0
5. Tiempo en que se recupera la inversión	2	5. ¿Cuán comprometida con el sistema está la administración del área usuaria?	5
Menos de 12 meses	1	Un tanto renuente o no se sabe	3
De 12 a 24 meses	2	Adecuada	2
Más de 24 meses	3	Muy entusiasta	1
Evaluación del riesgo por ESTRUCTURA	factor x peso riesgo		
1. ¿Cuál es la calificación general de la estructura predeterminada del sistema?	3	6. ¿Se ha establecido un equipo de trabajo en el cual participan los	5

usuarios y los informáticos?			Almacenamiento externo	1	
No	3		Terminales	2	
Hay representante del usuario, a tiempo parcial	1		Minicomputadores o micros	3	
Hay representante del usuario, a tiempo completo	0		Redes de computadores o telecomunicación	3	
7. ¿Hay algún otro sistema del cual depende el nuevo sistema?	6		4. ¿Cuántos proveedores ofrecen el hardware?		3
No	0		Uno	0	
Sí, pero ese proyecto tiene bajo riesgo	1		Dos	1	
Sí, pero ese proyecto tiene alto riesgo	3		Tres o más	3	
Evaluación del riesgo por TECNOLOGIA	factor x peso riesgo		5. ¿Es una aplicación distribuida?		1
1. ¿Qué tipo de hardware adicional se requiere?	1		No	0	
(suma)			Dos áreas geográficas	1	
Ninguno	0		Tres o más áreas geográficas	2	
Procesador	1		6. ¿Conoce el equipo del proyecto el siguiente software de base?		5
Almacenamiento externo	1		(suma)		
Terminales	2		Lenguaje de programación	3	
Minicomputadores o micros	3		Sistema administrador de bases de datos	3	
Redes de computadores o telecomunicación	3		Comunicación de datos	3	
2. ¿Cuál hardware es nuevo en la empresa?	4		Otros (especificar)	3	
(suma)			7. ¿Ha sido expuesto el usuario a la tecnología informática?		4
Ninguno	0		Primera vez que se expone	3	
Procesador	1		Exposición previa, conocimiento limitado	2	
Almacenamiento externo	1		Tine mucha capacidad	1	
Terminales	2		factor x peso riesgo		
Minicomputadores o micros	3		8. ¿Cuán conocedor del área propuesta de aplicación es el representante del usuario?		5
Redes de computadores o telecomunicación	3		Limitado	3	
3. ¿Cuál hardware es nuevo para el proveedor?	6		Entiende el concepto, sin experiencia	2	
(suma)			Ha participado en proyectos previos	1	
Ninguno	0		9. ¿Cuán conocedor del área		
Procesador	1				

propuesta de aplicación es el equipo técnico?	4
Limitado	3
Entiende el concepto, sin experiencia	2
Ha participado en proyectos previos	1

Nota: para los ítemes que indican (suma), se asignará el factor de cada subítem pertinente y luego la suma de ellos se multiplicará por el peso del ítem, para obtener su puntaje.

VIII Bibliografía Consultada.C1.

- Alavi84:** Alavi. **An Assessment of the Prototyping Approach to Information Systems Development.** Communications of the ACM, Vol. 27, No.6, jun 1984.
- Alavi85:** Alavi. **Application Prototyping** en Computer Programming Management. Pennsauken: Auerbach, 1985.
- Allen82:** Allen. **An unmanaged computer system can stop you dead.** Harvard Business Review, nov-dic 1982.
- Alter80:** Alter. **Decision Support Systems: current practices and continuing challenges.** Reading: Addison-Wesley, 1980.
- Andriole86:** Andriole (ed.). **Microcomputer decision support systems: design, implementation and evaluation.** Amsterdam: North-Holland; Wellesley: Q.E.D. Information Sciences, 1986.
- Badiru88:** Badiru. **Successful initiation of expert systems projects.** IEEE Transactions on Engineering Management, Vol.35, No.3, ago 1988.
- Beynon-Davies87:** Beynon-Davies. **Software engineering and knowledge engineering: unhappy bedfellows?** British Computer Society, Computer Bulletin, dic, 1987.
- Boar84:** Boar. **Application prototyping: a requirements definition strategy for the 80's.** New York: John Wiley and Sons, 1984.
- Boar86:** Boar. **Application prototyping: a life-cycle perspective.** Journal of Systems Management, Vol.37, No.2, feb 1986.
- Boehm81:** Boehm. **Software Engineering Economics.** Englewood Cliffs: Prentice-Hall, 1981.
- Boland87:** Boland; Hirschheim (eds.). **Critical issues in information systems research.** Chichester: John Wiley, 1987.
- Bondu89:** Bondu. **Planificación de Sistemas.** San José: Club de Investigación Tecnológica 1989.
- Brancheau87:** Brancheau; Leitheiser; Wetherbe. **Allocating Responsibility for End-User Computing Support** en Systems Development Management. Pennsauken: Auerbach, 1987.
- Brooks75:** Brooks. **The mythical man-month.** Reading: Addison-Wesley, 1975.
- Bruce82:** Bruce; Pederson. **The Software Development Project.** New York: John Wiley, 1982.

Buchanan80a: Buchanan; Linowes. **Understanding distributed data processing.** Harvard Business Review, jul-ago 1980.

Buchanan80b: Buchanan; Linowes. **Making distributed data processing work.** Harvard Business Review, set-oct 1980.

Buss83: Buss. **How to Rank Computer Projects.** Harvard Business Review, Ene-Feb 1983.

Campbell87: Campbell. **Reducing the Applications Backlog with 4GLs** en Computer Programming Management. Pennsauken: Auerbach, 1987.

Christiansen87: Christiansen. **Artificial expertise.** IEEE Spectrum, Vol.24, No.1, ene 1987.

Clark86: Clark. **Guidelines for Software Package Selection** en Computer Programming Management. Pennsauken: Auerbach, 1986.

Cordell85: Cordell. **The uneasy eighties: the transition to an information society.** Ottawa: Science Council of Canada, 1985.

Couger86: Couger. **E Pluribus Computum.** Harvard Business Review, set-oct 1986.

Cupello88: Cupello; Mishelevich. **Managing prototype knowledge/expert system projects.** Communications of the ACM, Vol.31, No.5, may 1988.

Daly79: Daly. **Organizing for succesful software development.** Datamation, dic 1979.

Davis78: Davis; Lawrence. **Problems of matrix organizations.** Harvard Business Review, Vol. 56, No.3, may-jun 1978.

De Marco82: De Marco. **Controlling Software Projects: measurement, management and estimation.** New York: Yourdon Press, 1982.

Dennis87: Dennis; Burns; Gallupe. **Phased Design: A Mixed Methodology for Application System Development.** Data Base, Vol.18, No.4, verano 1987.

Doll87: Doll; Torkzadeh. **The relationship of MIS steering committees to size of firm and formalization of MIS planning.** Communications of the ACM, No. 11, 1987.

Eliason87: Eliason. **Systems Development: analysis, design, implementation.** Boston: Little-Brown, 1987.

Fairley85: Fairley. **Software Engineering Concepts.** New York: McGraw-Hill, 1985.

Gilb88: Gilb; Finzli. **Principles of Software Engineering Management.** Reading: Addison-Wesley, 1988.

- Gilhooley85:** Gilhooley. **Purchasing Packaged Software** en Systems Development Management. Pennsauken: Auerbach, 1985.
- Gilhooley86:** Gilhooley. **The Impact of Prototyping on Systems Development** en Systems Development Management. Pennsauken: Auerbach, 1986.
- Gremillion83:** Gremillion; Pyburn. **Breaking the Systems Development Bottleneck.** Harvard Business Review, mar-abr 1983.
- Gutiérrez89:** Gutiérrez. **Sistemas Expertos.** San José: Club de Investigación Tecnológica, 1989.
- Hammond86:** Hammond. **Risk Assessment in Systems Development** en Systems Development Management. Pennsauken: Auerbach, 1986.
- Harmon85:** Harmon; King. **Expert systems: Artificial intelligence in business.** New York: John Wiley, 1985.
- Head82:** Head. **Strategic planning for information systems.** Wellesley: QED Information Sciences, 1982.
- Head84:** Head. **Planning techniques for systems management.** Wellesley: QED Information Sciences, 1984.
- Hirschheim85:** Hirschheim. **Office automation: a social and organizational perspective.** Chichester: John Wiley, 1985.
- Hogue87:** Hogue. **Developing Decision Support Systems** en Systems Development Management. Pennsauken: Auerbach, 1987.
- Huff85:** Huff. **Decision Support Systems** en Computer Programming Management. Pennsauken: Auerbach, 1985.
- Huff88:** Huff; Munro; Martin. **Growth Stages of End User Computing.** Communications of the ACM, Vol.31, No.5, may 1988.
- Jones86:** Jones. **Programming Productivity.** New York: McGraw-Hill, 1986.
- Keen84:** Keen; Woodman. **What to do with all those micros.** Harvard Business Review, set-oct 1984.
- Keen87:** Keen. **Managing Systems Development, 2nd. ed.** Chichester: John Wiley, 1987.
- King86:** King; Morgan. **Knowledge-based systems: an overview** en Systems Development Management. Pennsauken: Auerbach, 1986.

- Lansman88:** Lansman. **Improving Development with Application Generators** en Systems Development Management. Pennsauken: Auerbach, 1988.
- Larson87:** Larson; Gobeli. **Matrix Management: Contradictions and Insights.** California Management Review, Vol. 29, No. 4, verano 1987.
- Lee86:** Lee. **A Plan for Prototyping** en Computer Programming Management. Pennsauken: Auerbach, 1986.
- Lee88:** Lee. **The Problem with 4GLs** en Computer Programming Management. Pennsauken: Auerbach, 1988.
- Leonard-Barton87:** Leonard-Barton. **The case for integrative innovation: an expert system at Digital.** Sloan Management Review, Vol.29, No.1, otoño 1987.
- Leonard-Barton88:** Leonard-Barton; Sviokla. **Putting expert systems to work.** Harvard Business Review, mar-abr 1988.
- Ling86:** Ling. **Expert Systems for Business Applications.** Journal of Systems Management, Vol. 37, No.7, jul 1986.
- Lucas87:** Lucas. **Alternative Approaches to Traditional Systems Development** en Systems Development Management. Pennsauken: Auerbach, 1987.
- Lustman84:** Lustman. **Software Project Management.** Reston, Virginia: Reston Publishing, 1984.
- Martin83:** Martin; McClure. **Buying Software off the Rack.** Harvard Business Review, nov-dic 1983.
- McFarlan81:**McFarlan. **Portfolio approach to information systems.** Harvard Business Review, set-oct 1981.
- McFarlan83a:** McFarlan; McKenney; Pyburn. **The information archipelago - plotting a course.** Harvard Business Review, ene-feb 1983.
- McFarlan83b:** McFarlan; McKenney. **The information archipelago - governing the new world.** Harvard Business Review, jul-ago 1983.
- McKeen85:** McKeen; Guimaraes. **Selecting MIS projects by steering committee.** Communications of the ACM, Vol. 28, No. 12, 1985.
- McKenney82:** McKenney; McFarlan. **The information archipelago - maps and bridges.** Harvard Business Review, set-oct 1982.
- Metzger81:** Metzger. **Managing a Programming Project.** Englewood Cliffs: Prentice-Hall, 1981.

- Monge88:** Monge; Melegatti. **Usuarios de microcomputadoras y el centro de información, partes 1 y 2.** Mundo de la Computación, Nos. 14 y 15, 1988.
- Mumford87:** Mumford. **Managerial expert systems and organizational change: some critical research issues** en [Boland87].
- NCC87:** National Computer Centre; Department of Trade and Industry. **The Starts Guide, Vol. 1 & Vol. 2, 2nd. ed.** Manchester: NCC Publications, 1987.
- Nolan76:** Nolan. **Business needs a new breed of EDP manager.** Harvard Business Review, mar-abr 1976.
- Nolan82:** Nolan. **Managing information systems by committee.** Harvard Business Review, julio 1982.
- Olson86:** Olson; Turner. **Rethinking office automation.** Data Base, Vol.17, No.4, 1986.
- Page-Jones85:** Page-Jones. **Practical Project Management: restoring quality to DP projects and systems.** New York: Dorset House, 1985.
- Pliskin87:** Pliskin; Shoval. **End-user prototyping: sophisticated users supporting system development.** Data Base, Vol.18, No.4, verano 1987.
- Pressman87:** Pressman. **Software Engineering: a Practical Approach, 2nd. ed.** New York: McGraw-Hill, 1987.
- Price86:** Price. **Managing Computer Projects.** Chichester: John Wiley, 1986.
- Reck86:** Reck; Hall. **Designing Executive Information Systems** en Systems Development Management. Pennsauken: Auerbach, 1986.
- Reimann85:** Reimann; Waren. **User-oriented criteria for the selection of DSS software.** Communications of the ACM, Vol.28, No.2, feb 1985.
- Rosenau84:** Rosenau; Lewin. **Software Project Management: Step by Step.** Belmont, California: Wadsworth (Lifetime Learning Publications), 1984.
- Rubin86:** Rubin. **Save your information systems from the experts.** Harvard Business Review, julio 1976.
- Ryan86a:** Ryan. **Major Issues in End-User Computing** en Systems Development Management. Pennsauken: Auerbach, 1986.
- Ryan86b:** Ryan. **The Role of Systems Development in End-User Computing** en Systems Development Management. Pennsauken: Auerbach, 1986.

- Shere88:** Shere. **Software Engineering and Management.** Englewood Cliffs: Prentice-Hall, 1988.
- Slevin87:** Slevin; Pinto. **Balancing strategy and tactics in project implementation.** Sloan Management Review, Vol.29, No.1, otoño 1987.
- Sommerville85:** Sommerville. **Software Engineering, 2nd. ed.** Workingham: Addison-Wesley, 1985. (Traducido al español como Ingeniería de software).
- Sviokla86a:** Sviokla. **Business implications of knowledge-based systems - part 1.** Data Base, Vol. 17, No.4, verano 1986.
- Sviokla86b:** Sviokla. **Business implications of knowledge-based systems - part 2.** Data Base, Vol. 18, No.1, otoño 1986.
- Tangui85:** Tangui. **Elección y compra de software de gestión.** Bilbao: Deusto, 1985.
- Thayer80:** Thayer; Pyster; Wood. **The challenge of software engineering project management.** IEEE Computer, ago 1980. (También aparece en [Trejos84]).
- Thayer82:** Thayer; Pyster; Wood. **Validating solutions to major problems in software engineering project management.** IEEE Computer, ago 1982.
- Trejos84:** Trejos (ed.). **Software: filosofía e ingeniería.** Cartago: ITCR, 1984.
- Wilkinson87:** Wilkinson. **Organizing for project control** en Computer Programming Management. Pennsauken: Auerbach, 1987.
- Wilson88:** Wilson; Sifer. **Structured planning - project views.** Software Engineering Journal, jul 1988.
- Yourdon82:** Yourdon. **Managing the system life cycle:A software development methodology overview.** New York: Yourdon Press,1982

